Adaptive Mesh Refinement on Overlapping Grids

Bill Henshaw

Centre for Applied Scientific Computing,

Lawrence Livermore National Laboratory,

Livermore, CA, USA  94551

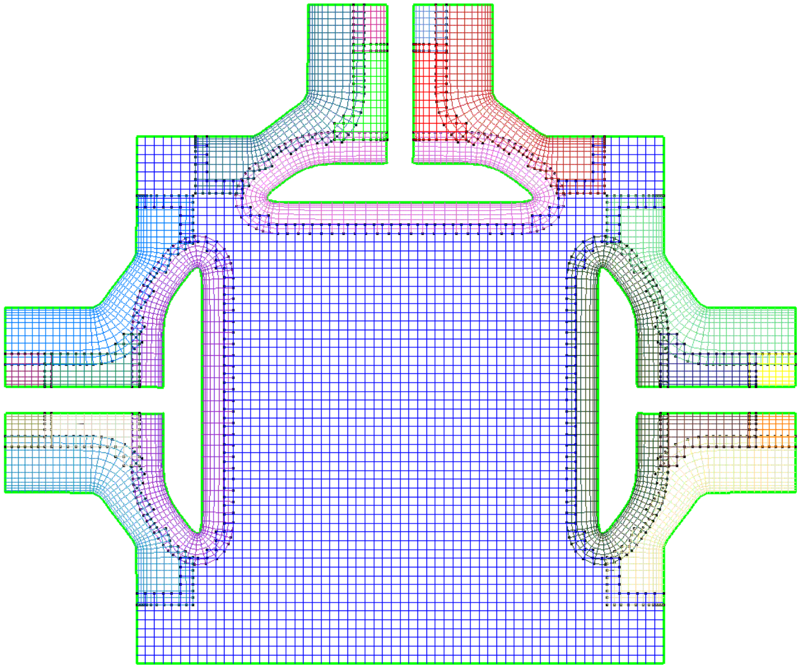
`www.llnl.gov/casc/Overture`

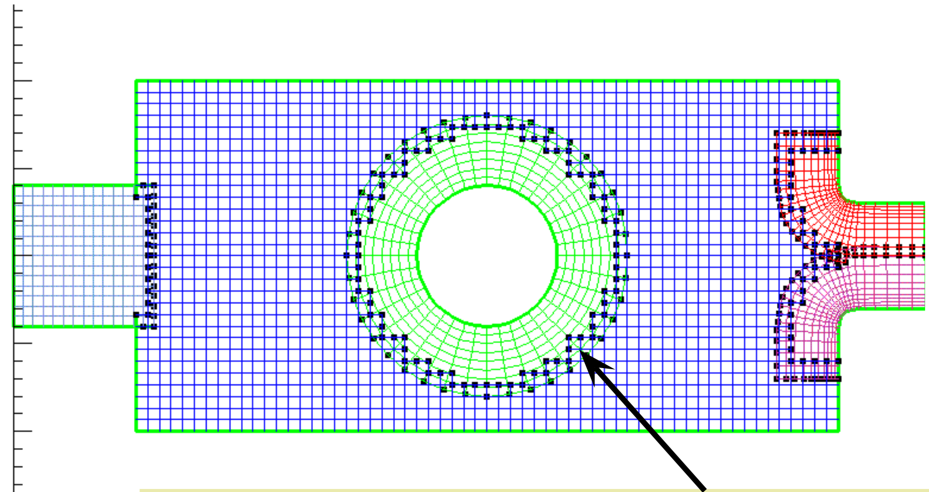
In collaboration with: Don Schwendeman (RPI)

Talk presented in Sweden  June 2004.

Talk summary

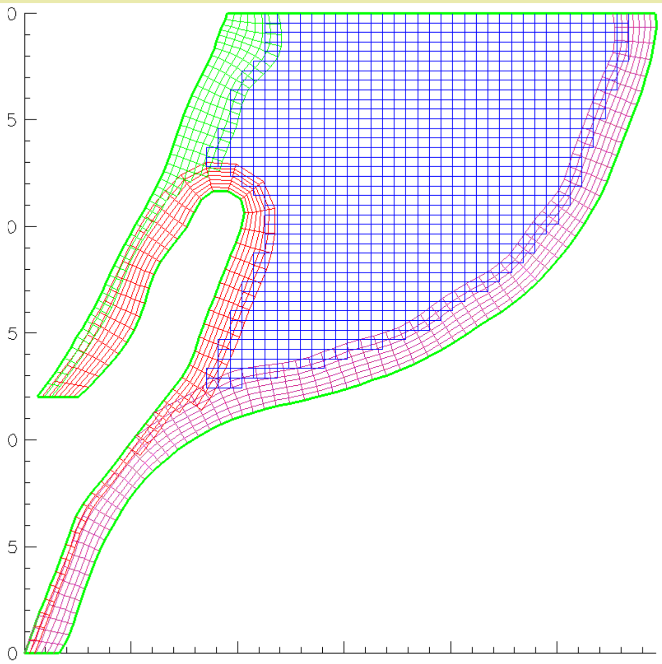
- Overview of the Overture framework, current projects.
- AMR on overlapping grids, existing capabilities.
- Some sample denonation computations.
- Current work: three-dimensions, moving geometry and AMR, parallel

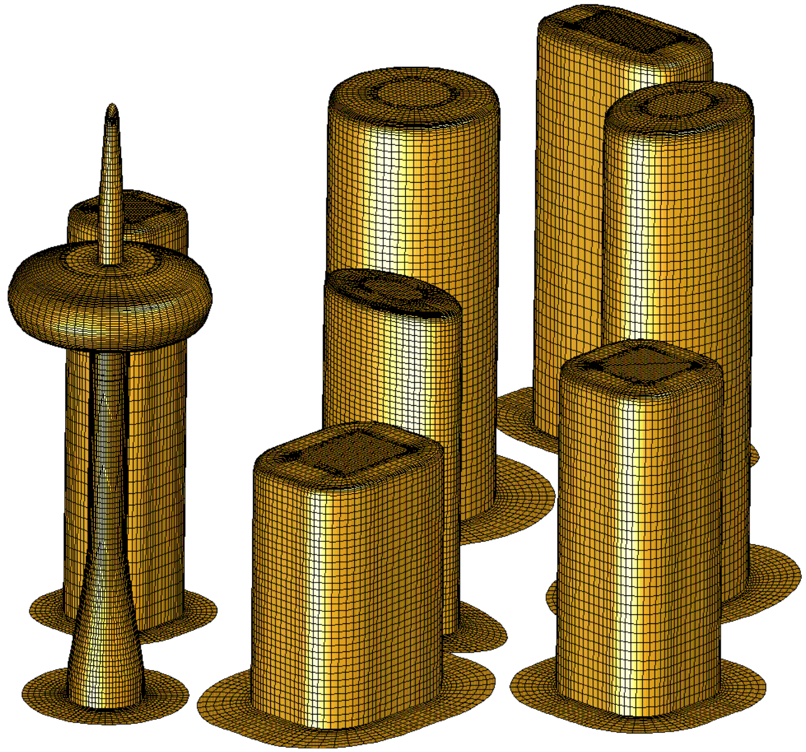


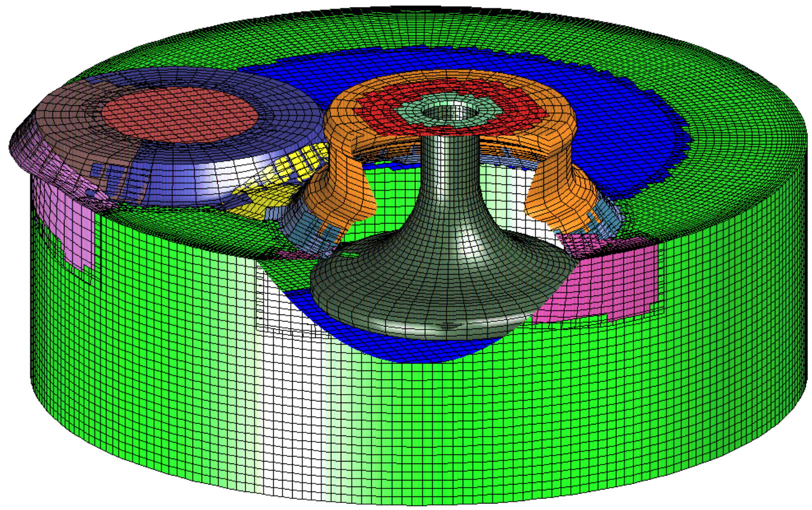


Solutions coupled by interpolation

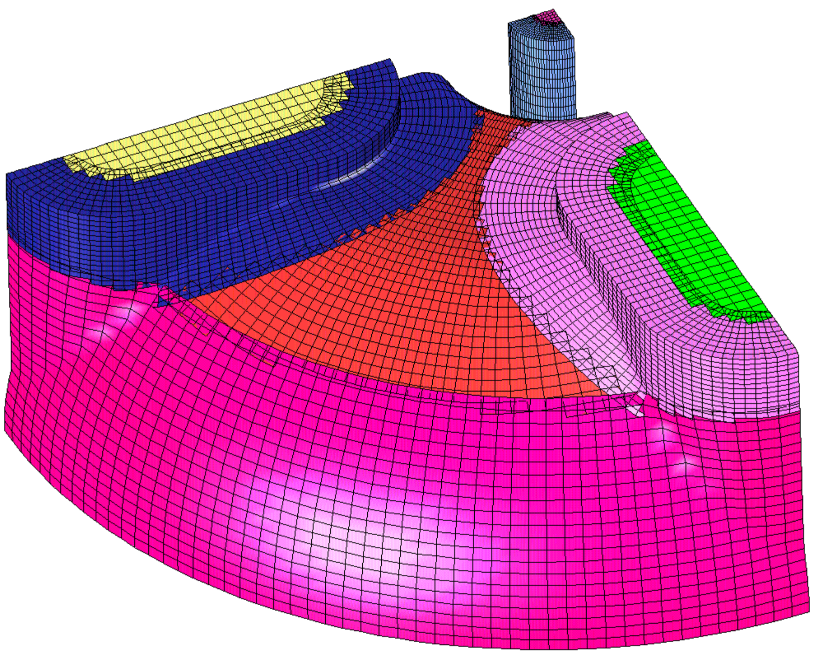
Sample 2D overlapping grids

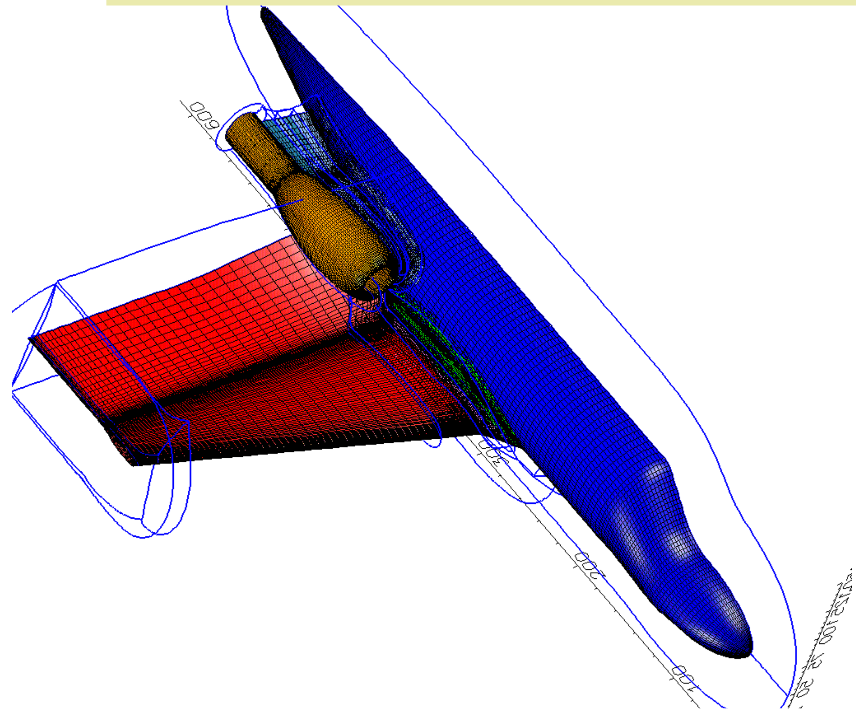


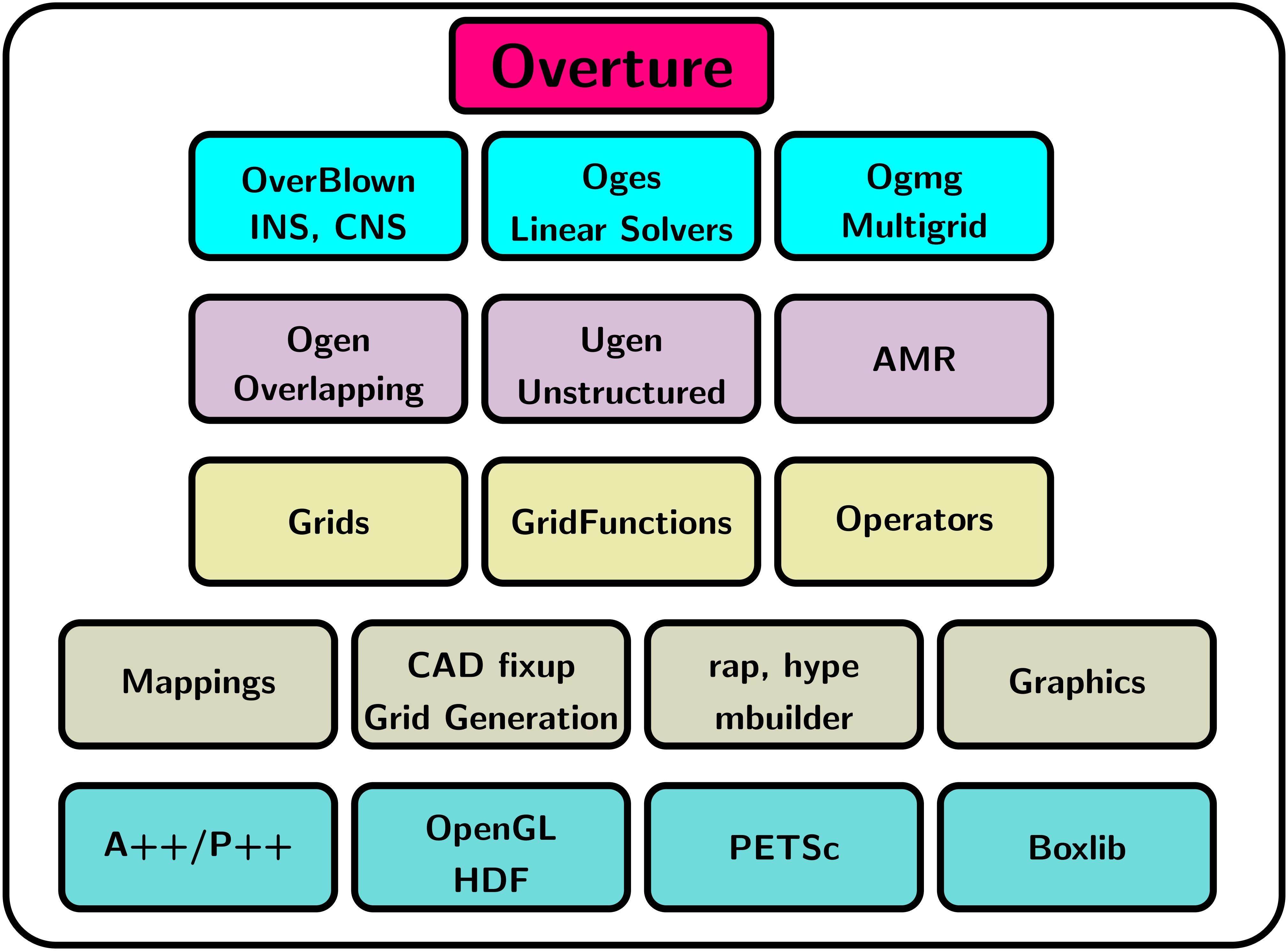




Sample 3D overlapping grids






Overture
OverBlown
INS, CNS
Oges
Linear Solvers
Ogmg
Multigrid
Ogen
Overlapping
Ugen
Unstructured
AMR
Grids
GridFunctions
Operators
Mappings
CAD fixup
Grid Generation
rap, hype
mbuilder
Graphics
A++/P++
OpenGL
HDF
PETSc
Boxlib

Overture supports a high-level C++ interface (but is built mainly upon Fortran kernels):

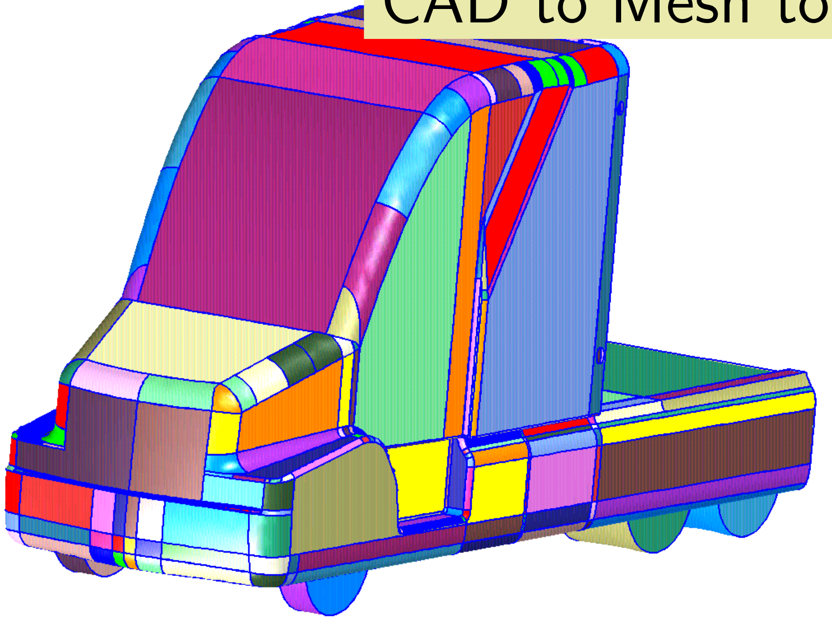
Solve $u_t + au_x + bu_y = \nu(u_{xx} + u_{yy})$

```
CompositeGrid cg;  // create a composite grid
getFromADataBaseFile(cg,"myGrid.hdf");
floatCompositeGridFunction u(cg);  // create a grid function
u=1.;
CompositeGridOperators op(cg);  // operators
u.setOperators(op);
float t=0, dt=.005, a=1., b=1., nu=.1;
for( int step=0; step<100; step++ )
{
  u+=dt*( -a*u.x()-b*u.y()+nu*(u.xx()+u.yy()) ); // forward Euler
  t+=dt;
  u.interpolate();
  u.applyBoundaryCondition(0,dirichlet,allBoundaries,0.);
  u.finishBoundaryConditions();
}
```

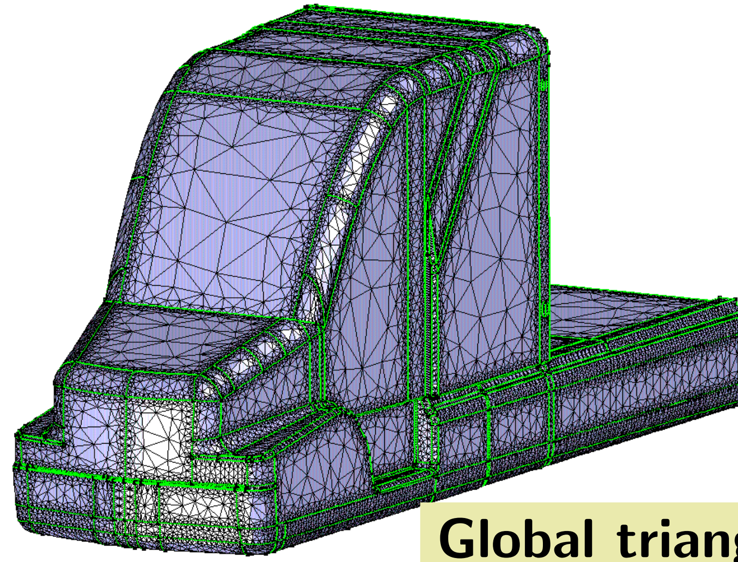
Current Projects with Overture

- Hybrid (unstructured) grid generation and algorithms (Kyle Chand)
  - the overlap region is replaced by an unstructured grid (advancing front algorithm).
  - a stabilized DSI scheme for Maxwell's equations on hybrid grids.
- Deforming boundaries in incompressible flow (Petri Fast).
- Multigrid solvers for elliptic problems on overlapping grids.
  - robust coarsening and adaptive smoothing techniques.
  - second- and fourth-order accurate, Dirichlet and Neumann boundary conditions.
- Incompressible Navier-Stokes solvers for overlapping grids.
  - fourth-order accurate time accurate solver.
  - line-implicit pseudo-steady state solver.
- High speed reactive flow and adaptive mesh refinement (with Don Schwendeman (RPI))
- An overlapping grid solver for the time dependent Maxwell's equations.
  - fourth-order accurate, parallel

CAD to Mesh to Solution with Overture

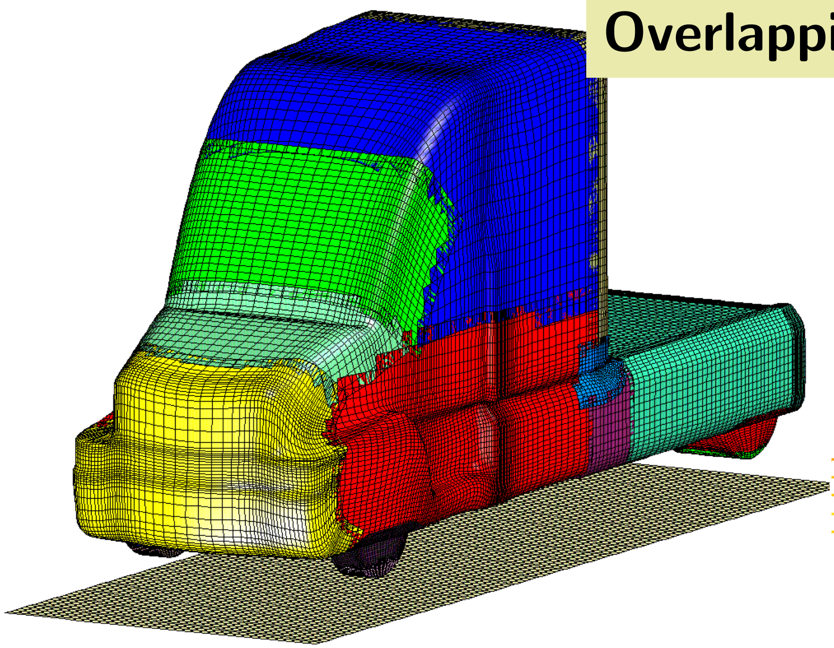


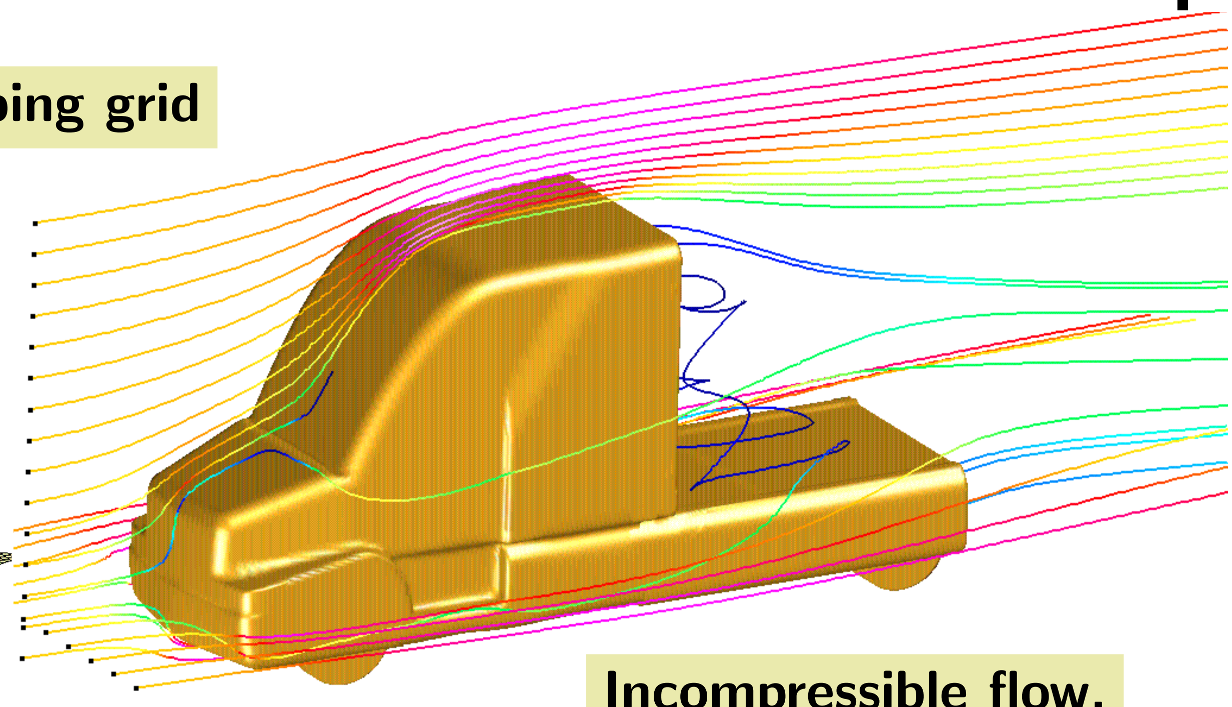
Cad fixup



Global triangulation

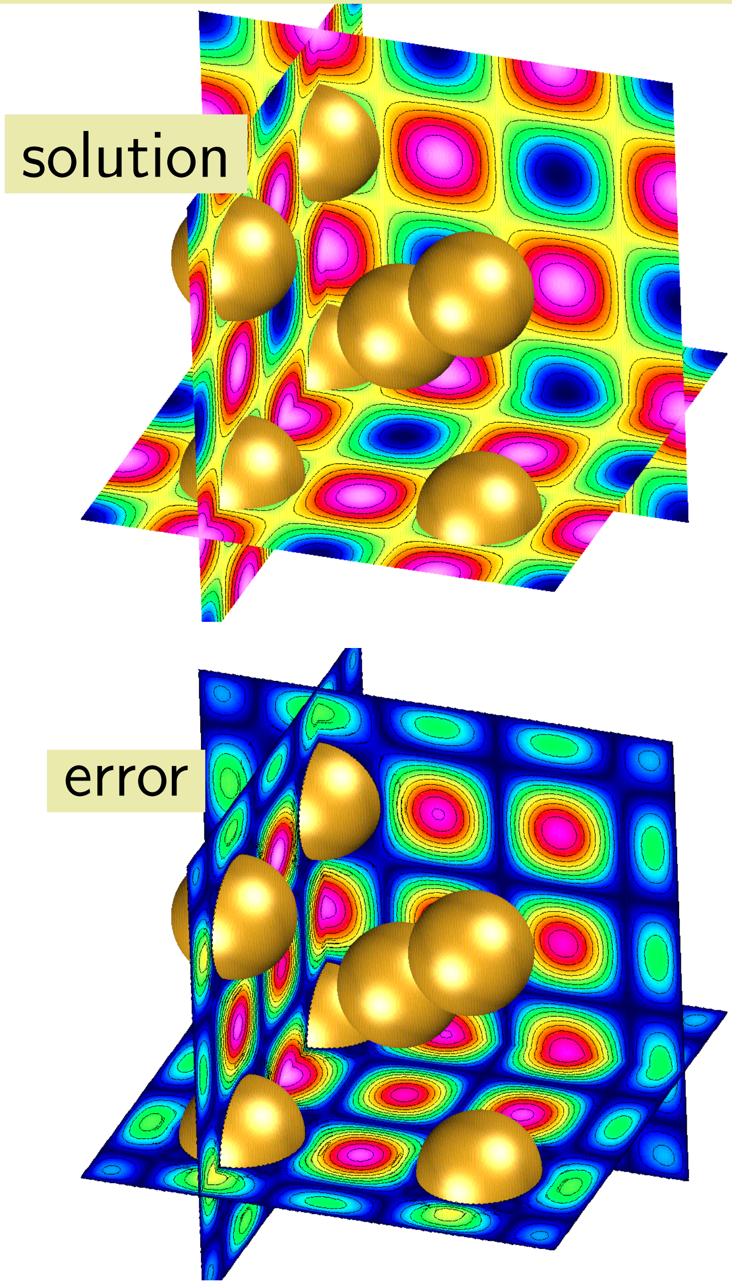
Overlapping grid

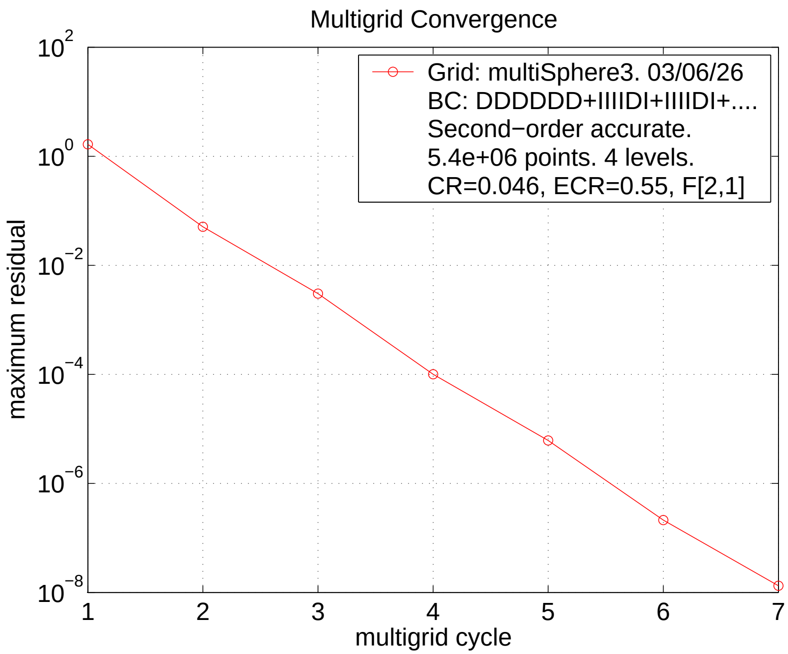


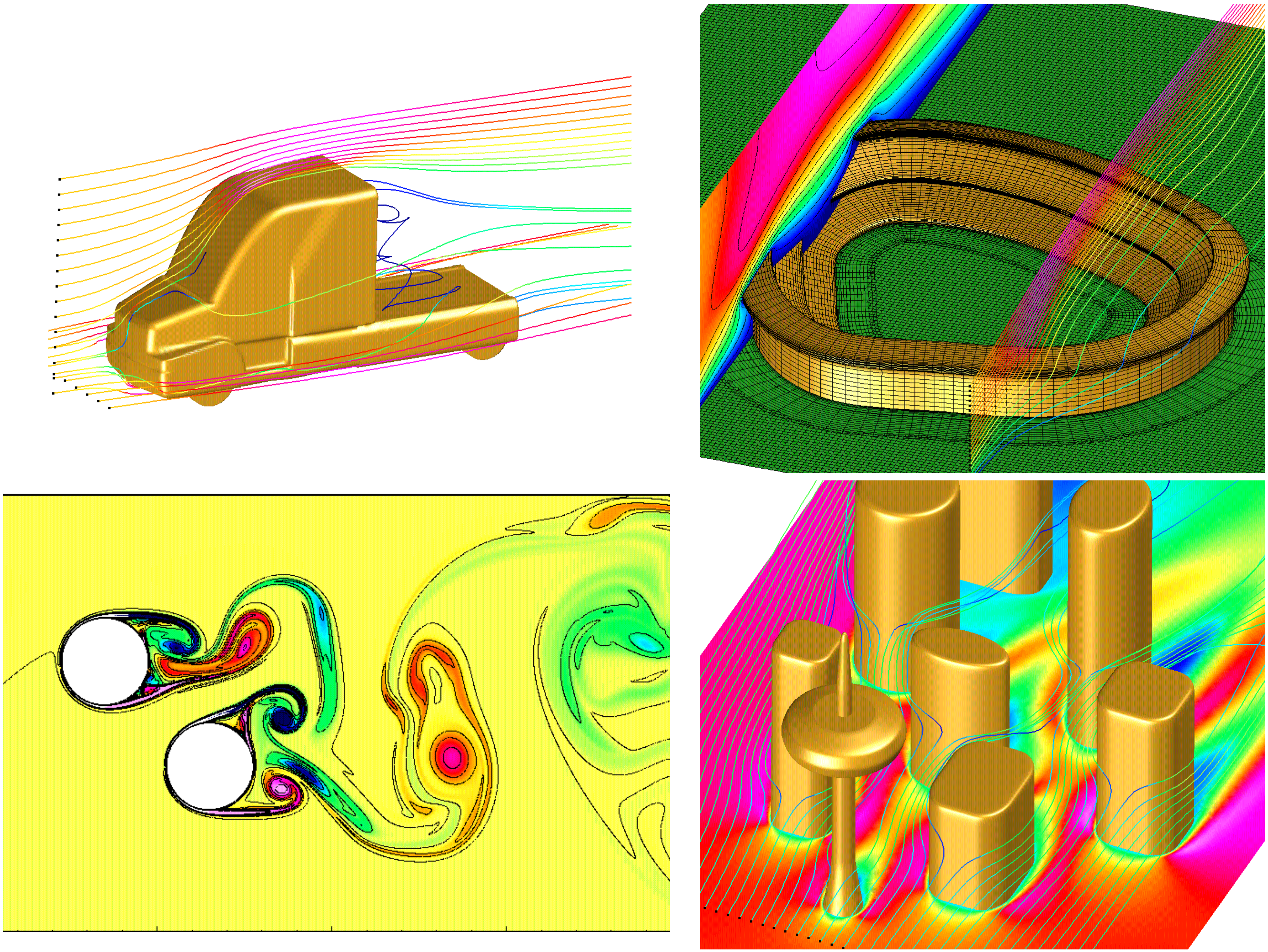


Incompressible flow.

Multigrid solution to Poisson's equation, 5.4 million grid points

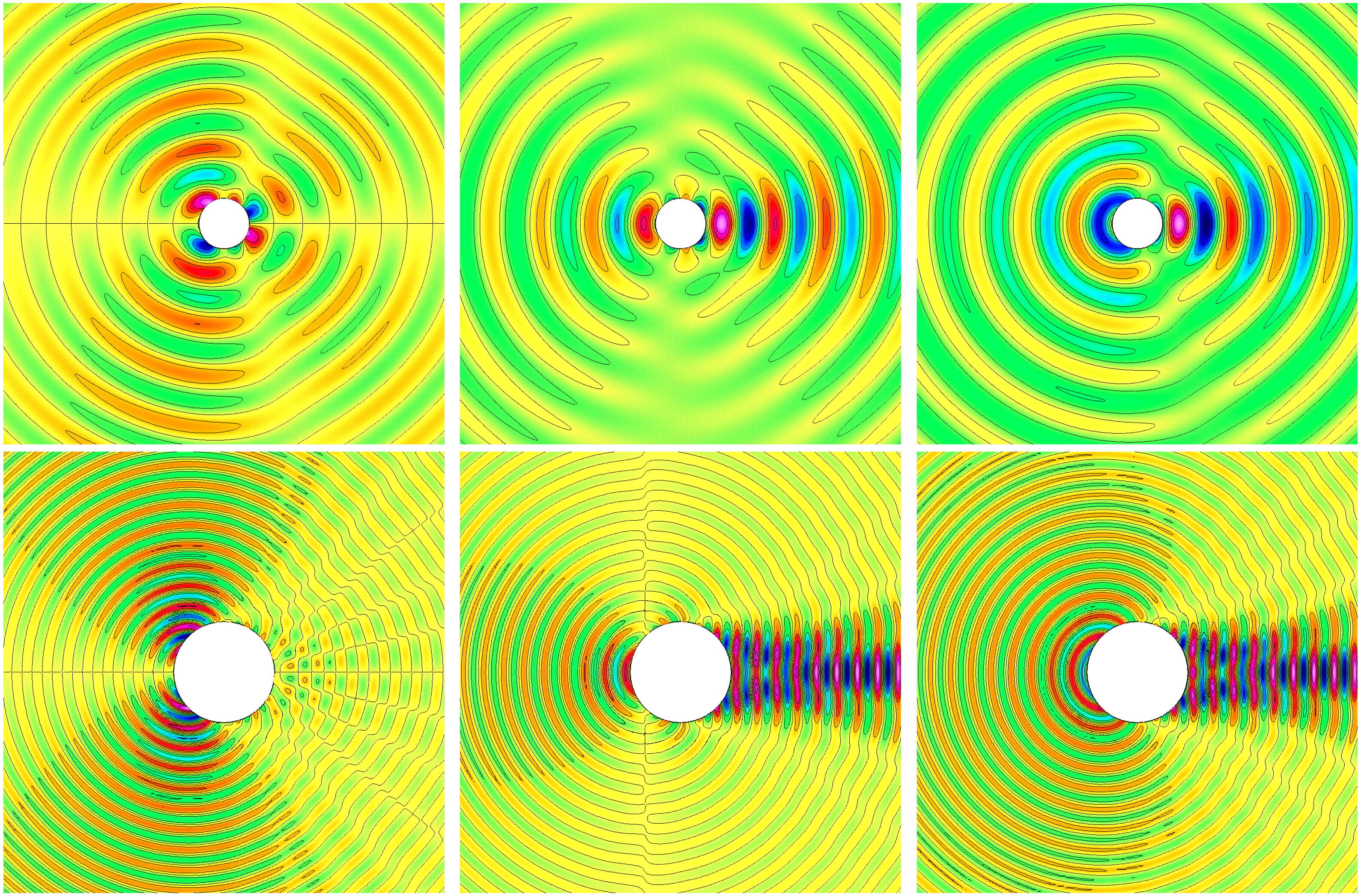






Incompressible flow computations with OverBlown.

A Parallel 4th-order accurate solver for the time-dependent Maxwell equations



Scattering of a plane wave by a cylinder. Top: scattered field E_x, E_y and H_z for $ka = 1/2$.
Bottom: scattered field E_x, E_y and H_z for $ka = 5/2$

Block structured Adaptive Mesh Refinement

◇ Initially developed by Berger and Oliger (JCP 1984)

◇ Extensions to the Euler equations by Berger and Colella (JCP 1989)

◇ AMR and overlapping grids considered by Brislawn, Brown, Chesshire and Saltzman (1995), and Boden and Toro (1997)

◇ AMR in Overture has contributions from Brown, Philip and Quinlan.

Some Structured AMR frameworks

◇ AMRCLAW (LeVeque and Berger)

◇ Amrita (Quirk)

◇ Boxlib (Bell et.al., LBNL)

◇ Chombo (Colella et.al., LBNL)

◇ GrACE (Parashar)

◇ PARAMESH (NASA Goddard Space Flight Center)

◇ SAMRAI (Hornung et.al. LLNL)

Reference Tomasz Plewa's AMR page `http://flash.uchicago.edu/~tomek/AMR`

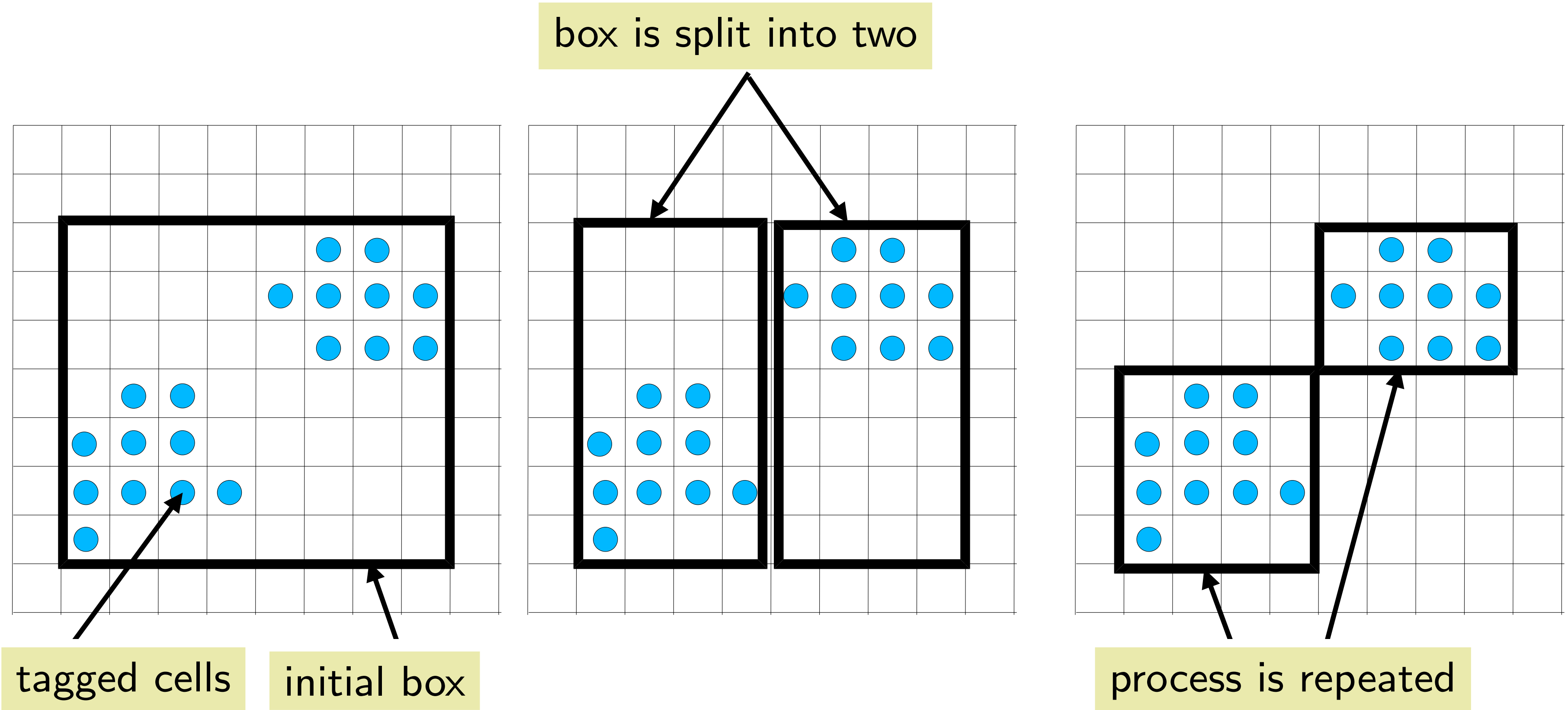
AMR on overlapping grids

Currently adding AMR capabilities to Overture for overlapping grids.

Some of the issues that need to be addressed

- multiple base grids.
- efficient handling of refinement grids on curvilinear grids.
- support for higher order accurate methods (fourth-order, sixth-order,...)
- updating refinement grids that meet at the overlapping grid boundaries.
- retaining the efficiency of cartesian grids.
- saving and reading solutions and grids from a data base file in an efficient manner (e.g. for post-processing and restarts).
- graphics.

AMR regridding algorithm (Berger-Rigoutsos)



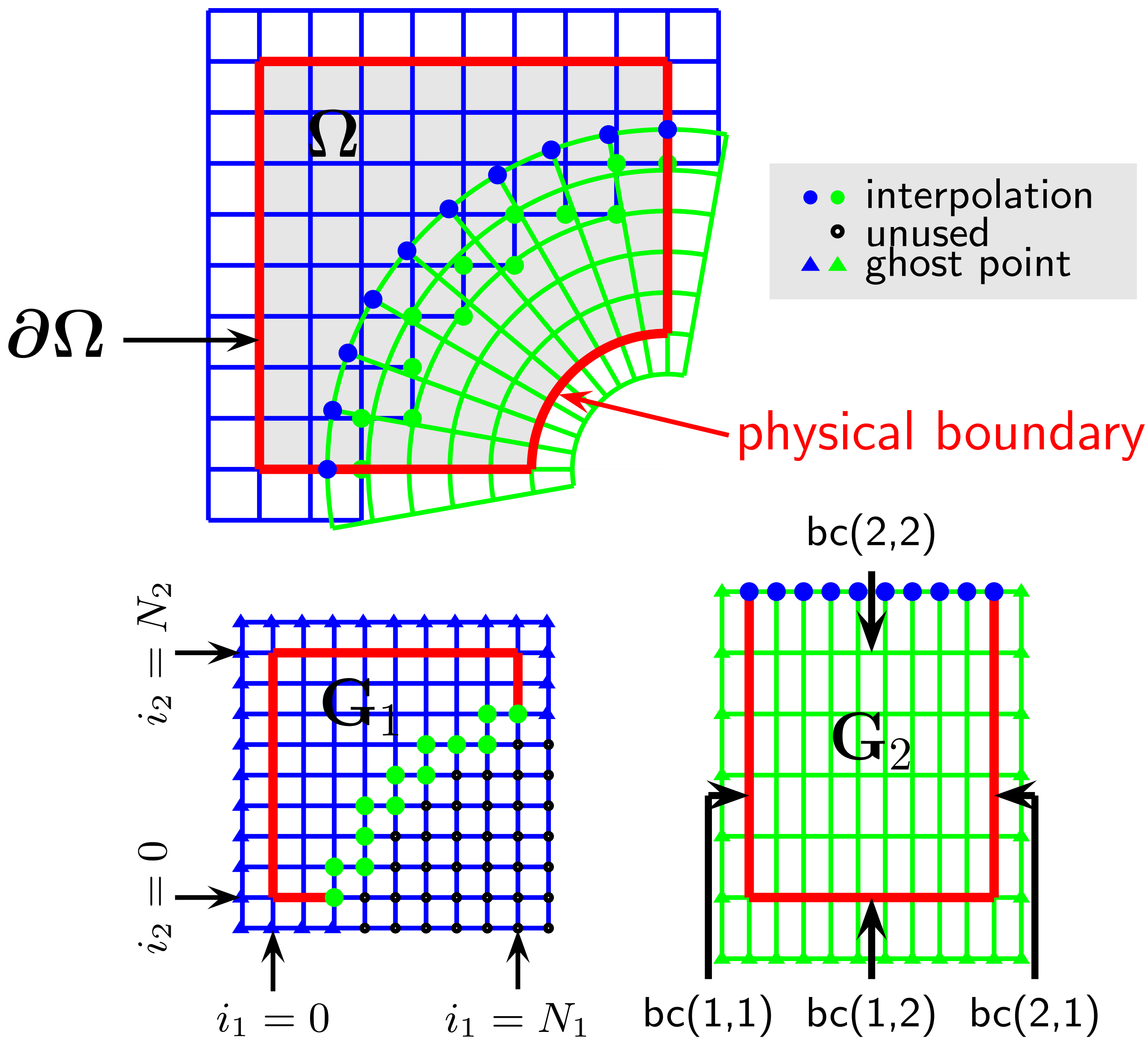
(1) tag cells where refinement is needed

(2) create a box to enclose tagged cells

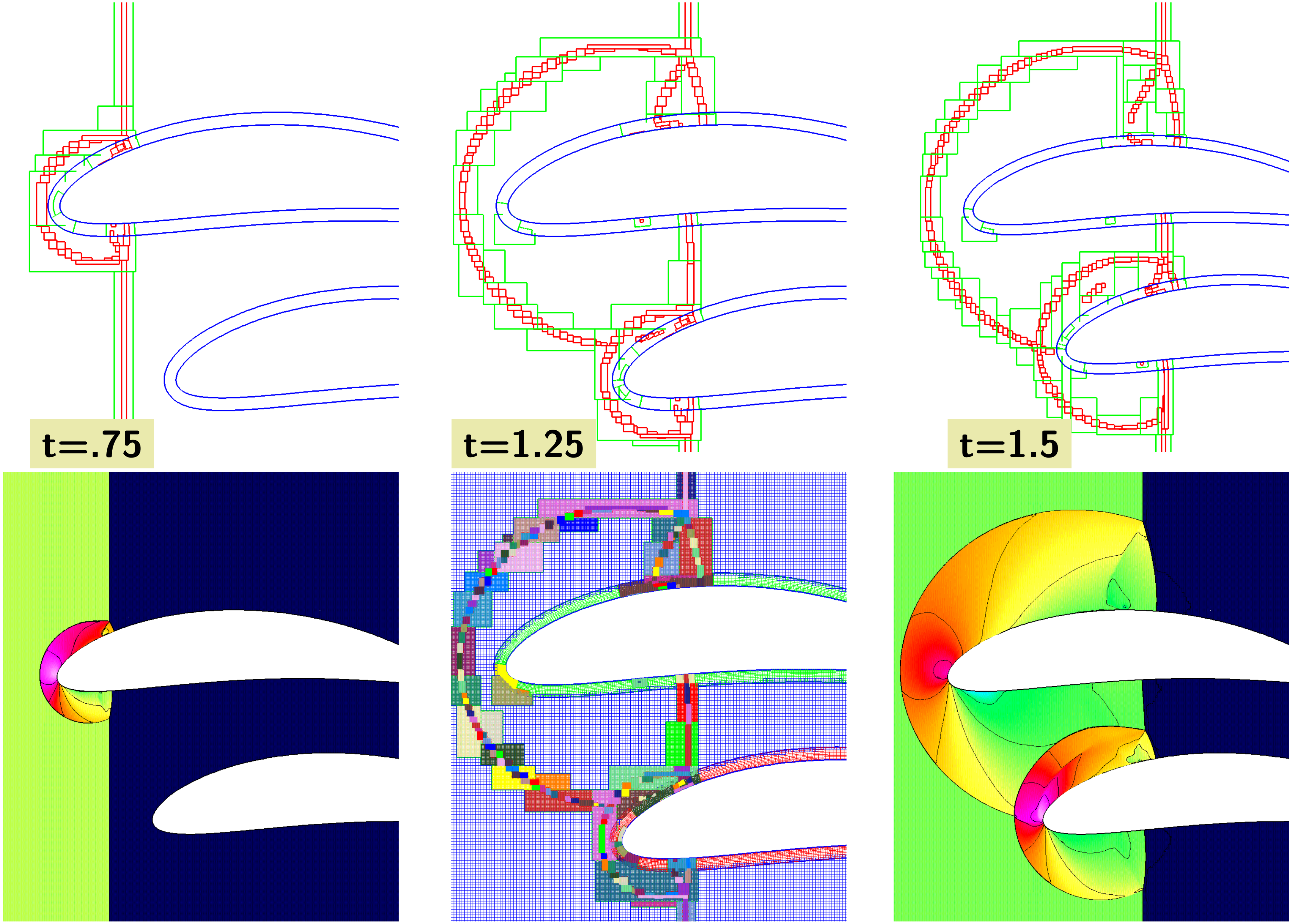
(3) split the box along its long direction based on a histogram of tagged cells

(4) fit new boxes to each split box and repeat the steps as needed.

Components of an Overlapping Grid



Adaptive overlapping grids

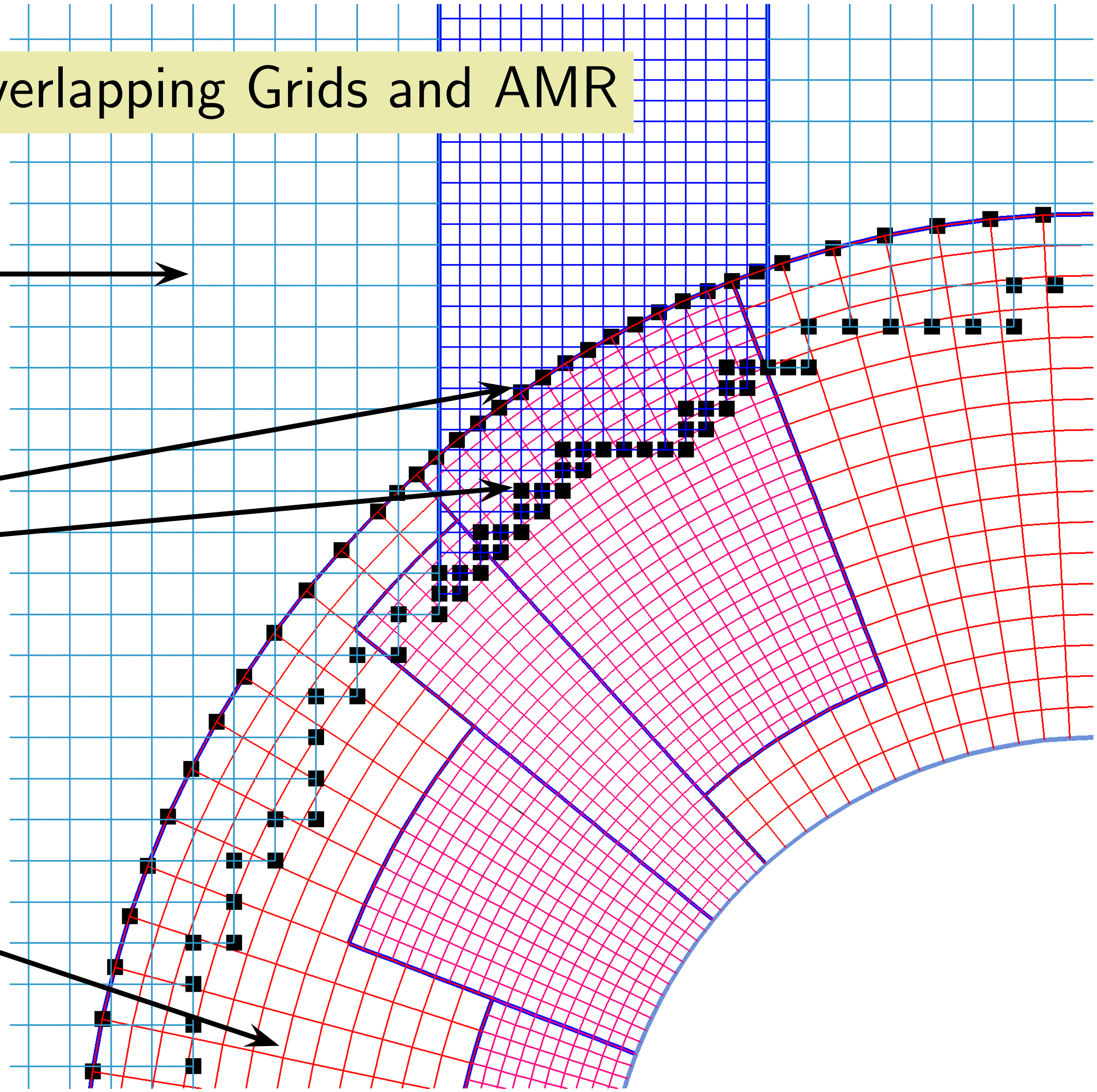


Overlapping Grids and AMR

Component grid 1,
base grid 1

Refinement grids
interpolate from
refinements of a different
base grid

Component grid 2,
base grid 2



The basic AMR time stepping algorithm

PDEsolve$(\mathcal{G}, t_{\text{final}})$

{

$t := 0;\ n := 0;$

$\mathbf{u}_{\mathbf{i}}^n :=$ **applyInitialCondition**$(\mathcal{G});$

while $t < t_{\text{final}}$

if $(n$ **mod** $n_{\text{regrid}} == 0)$

$e_{\mathbf{i}} :=$ **estimateError**$(\mathcal{G}, \mathbf{u}_{\mathbf{i}}^n);$

$\mathcal{G}^* :=$ **regrid**$(\mathcal{G}, e_{\mathbf{i}});$

$\mathbf{u}_{\mathbf{i}}^* :=$ **interpolateToNewGrid**$(\mathbf{u}_{\mathbf{i}}^n, \mathcal{G}, \mathcal{G}^*);$

$\mathcal{G} := \mathcal{G}^*;\ \mathbf{u}_{\mathbf{i}}^n := \mathbf{u}_{\mathbf{i}}^*;$

end

$\Delta t :=$ **computeTimeStep**$(\mathcal{G}, \mathbf{u}_{\mathbf{i}}^n);$

$\mathbf{u}_{\mathbf{i}}^{n+1} :=$ **timeStep**$(\mathcal{G}, \mathbf{u}_{\mathbf{i}}^n, \Delta t);$

$t := t + \Delta t;\ n := n + 1;$

interpolate$(\mathcal{G}, \mathbf{u}_{\mathbf{i}}^n);$

applyBoundaryConditions$(\mathcal{G}, \mathbf{u}_{\mathbf{i}}^n, t);$

end

}

The basic components of AMR in Overture

Error estimation

- standard error estimators based on first and second differences.
- smoothing of the error and propagation across overlapping grid boundaries.

Regridding

- generation of aligned AMR grids using the Berger-Rigoutsos algorithm.
- Boxlib is used for domain calculus (e.g. intersecting two boxes).
- updating the overlapping grid interpolation points on AMR refinement grids.

Interpolation

- fine to coarse and coarse to fine interpolation
- support for *any* refinement ratio (1,2,3,4,...) and *any* order of accuracy.
- high level functions to interpolate the solution from one AMR overlapping grid to another AMR overlapping grid.
- functions to update all AMR ghost points and hidden coarse grid points on an AMR overlapping grid.

AMR on overlapping grids - high-level objects

Grids and solutions on the overlapping grid AMR hierarchy are represented as

```
CompositeGrid cg; // (derived from a GridCollection)
realCompositeGridFunction u(cg,all,all,all,3); // (3 components)
```

Individual grids can be accessed as

```
MappedGrid & mg = cg[grid];
realMappedGridFunction & umg = u[grid];
```

All grids on a refinement level can be accessed as

```
GridCollection & rl = cg.refinementLevel[level];
realGridCollectionFunction & url = u.refinementLevel[level];
```

amrHype: Solve a convection diffusion equation with AMR

- a small code demonstrating the use of AMR with Overture.
- uses the *method of analytic solutions* for testing accuracy.

Sample uses of Overture AMR functions:

```
CompositeGrid cg,cgNew;
realCompositeGridFunction u,error,uNew;
ErrorEstimator errorEstimator;
    errorEstimator.computeAndSmoothErrorFunction(u,error);
Regrid regrid;
    regrid.regrid(cg,cgNew, error, errorThreshold );
Ogen ogen;
    ogen.updateRefinement(cgNew);
uNew.updateToMatchGrid(cgNew);
InterpolateRefinements interp;
    interp.interpolateRefinements( u,uNew );
```

Testing using the method of analytic solutions

The usefulness of this technique cannot be overstated.

Given a PDE boundary value problem

$$L(u_t, u_x, u_y, \ldots) = F(\mathbf{x}, t)$$

one can create an *exact* solution, $U(\mathbf{x}, t)$ by choosing

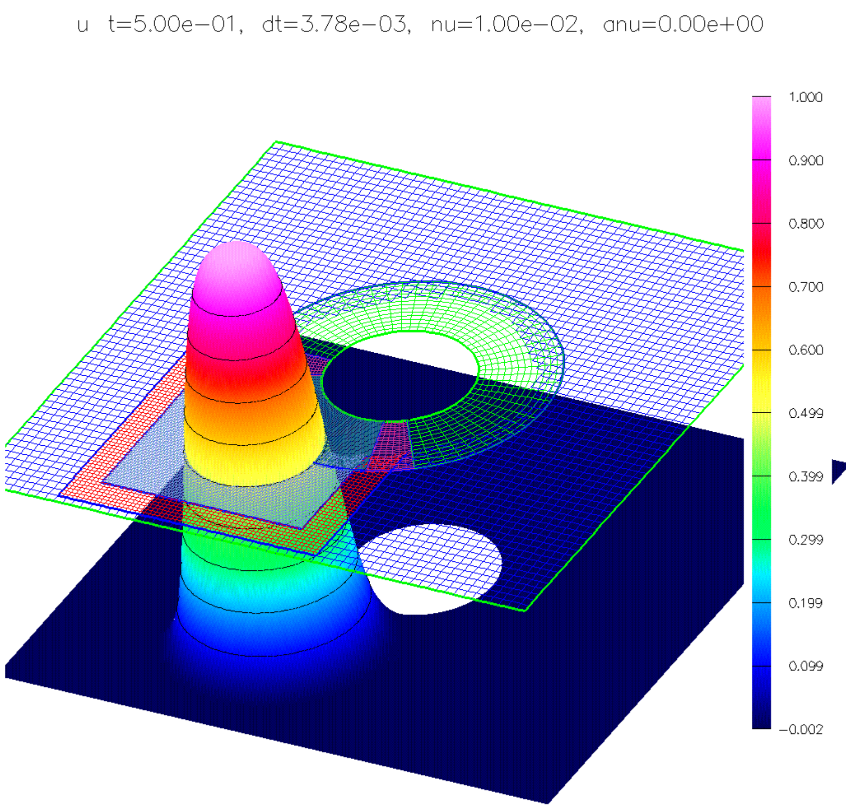
$$F(\mathbf{x}, t) = L(U_t, U_x, U_y, \ldots)$$

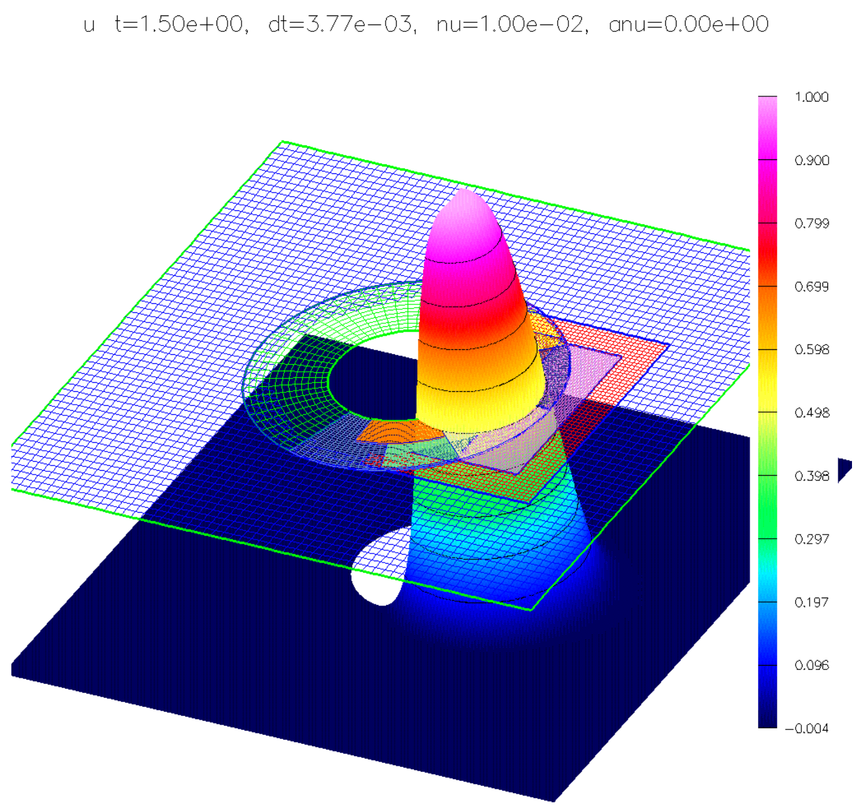
The Overture OGFunction class defines a variety of exact solutions and their derivatives to support the method of analytic solutions. For example one could define a polynomial, trigonometric polynomial, or *pulse function*

$$U(\mathbf{x}, t) = (x^2 + 2xy + y^2 + z^2)(1 + \frac{1}{2}t + \frac{1}{3}t^2)$$
$$U(\mathbf{x}, t) = \cos(\pi\omega x)\cos(\pi\omega y)\cos(\pi\omega z)\cos(\omega_3 \pi t)$$
$$U(\mathbf{x}, t) = a_0 \exp(-a_1 \|\mathbf{x} - \mathbf{b}(t)\|^{2p}) \ , \quad \mathbf{b}(t) = \mathbf{c}_0 + \mathbf{v}t$$

The polynomial solution is particularly useful since this solution is often an exact solution to the discrete equations on rectangular grids. The pulse function is good for AMR.

amrHype: Solve a convection diffusion equation with AMR





Traveling pulse analytic solution

High Speed Reactive Flow Project

$\diamondsuit$ Develop software tools for the numerical solution of the reactive Euler equations with a general equation of state and various reaction rate models.

- $\diamondsuit$ AMR to resolve shocks/detonations with numerical efficiency
- $\diamondsuit$ Sub-CFL time step resolution for fast chemical reactions
- $\diamondsuit$ Overlapping grids to handle complex two- and three-dimensional geometries.

$\diamondsuit$ Parallel processing (in progress)

$\diamondsuit$ Study detonation dynamics in homogeneous and heterogeneous explosives. For example:

- $\diamondsuit$ Explore paths to detonation of reactive samples at critical conditions subjext to small initial non-uniformities
- $\diamondsuit$ Explore detonation/confinement interactions with applications to detonation diffraction and failure

$\diamondsuit$ explore features and limitations of existing models (e.g. ignition-and-growth), and explore new models (e.g. multiphase)

$\diamondsuit$ Reference *An Adaptive Numerical Scheme for High-Speed Reactive Flow on Overlapping Grids*, JCP vol. 191, 2003.

Numerical Method

Summary:

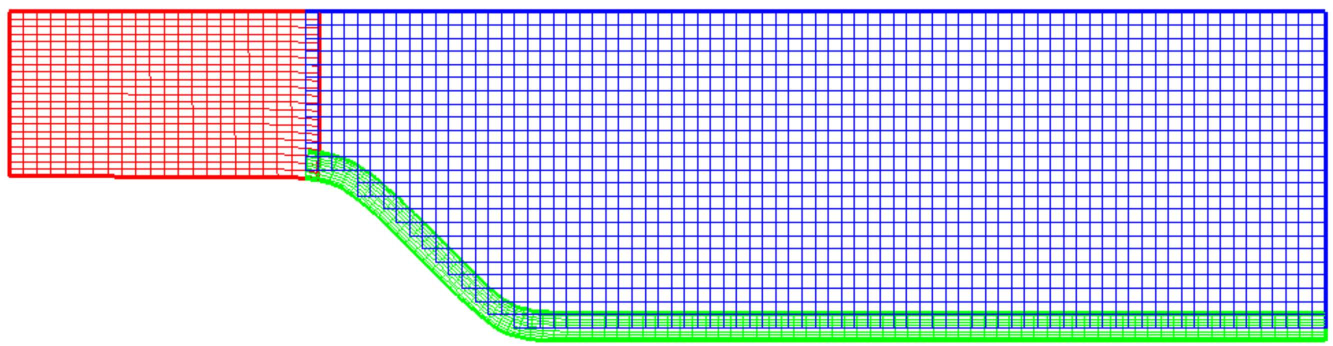
◇ Domain is covered by a collection of overlapping curvilinear grids.

◇ Solution is advanced in time on each component grid according to a second-order accurate, shock-capturing, Godunov-type scheme.

◇ Source term is handled with an error-control algorithm allowing sub-CFL time step resolution

◇ Interpolation is used near the overlap boundaries to communicate the solution between component grids.

◇ AMR is used to resolve the fine-scale structures.



A sample overlapping grid consisting of an inlet grid,
a background grid and a boundary grid

Reactive Euler Equations

Governing equations (2-D):

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x + \mathbf{g}(\mathbf{u})_y = h(\mathbf{u})$$

where

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \\ \rho\lambda \end{bmatrix} \quad \mathbf{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho E + p) \\ \rho u\lambda \end{bmatrix} \quad \mathbf{g} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \\ \rho v\lambda \end{bmatrix} \quad \mathbf{h} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \rho R \end{bmatrix}$$

Variables:

$\rho =$ density $\qquad (u, v) =$ velocity

$p =$ pressure $\qquad E =$ total energy

$\lambda = n$ mass fractions $\qquad R = n$ reaction rates

$$E = e + \frac{1}{2}(u^2 + v^2)$$

where

$$e = e(\rho, p, \lambda) = \begin{cases} \text{internal energy per unit mass} \\ \text{(as specified by an equation of state)} \end{cases}$$

Reaction and equation-of-state models

One-step:

$$\mathcal{F} \xrightarrow{k_C} \mathcal{P}, \qquad k_C = \sigma \exp\left(\frac{1}{\epsilon}\Big(\frac{1}{T_C} - \frac{1}{T}\Big)\right)$$

Variables:

$\mathcal{F}$ = fuel $\qquad \mathcal{P}$ = product

T = temperature, $= p/\rho \qquad T_C$ = cross-over temperature

σ = prefactor (sets time scale) $\qquad \epsilon$ = reciprocal activation energy, $\ll 1$

Reaction rate:

$R = (1-\lambda)k_C, \quad \lambda$ = mass fraction of product

Equation of state:

$$e = \frac{p}{(\gamma-1)\rho} + \lambda Q$$

where

γ = ratio of specific heats

Q = heat release < 0 (exothermic)

Chain-Branching: (Kapila)

$$\mathcal{F} \xrightarrow{k_I} \mathcal{Y}, \qquad k_I = \sigma \exp\left(\frac{1}{\epsilon_I}\left(\frac{1}{T_I} - \frac{1}{T}\right)\right) \quad \text{(initiation)}$$

$$\mathcal{F} + \mathcal{Y} \xrightarrow{k_B} \in\mathcal{Y}, \qquad k_B = \sigma \exp\left(\frac{1}{\epsilon_B}\left(\frac{1}{T_B} - \frac{1}{T}\right)\right) \quad \text{(branching)}$$

$$\mathcal{Y} \xrightarrow{k_C} \mathcal{P}, \qquad k_C = 1 \quad \text{(sets time scale)} \quad \text{(completion)}$$

where

$$\mathcal{F}, \mathcal{Y}, \mathcal{P} = \text{fuel, radical, product}$$

$$T_I, T_B = \text{cross-over temperatures}$$

$$\epsilon_I, \epsilon_B = \text{reciprocal activation energies}$$

Reaction rates:

$$R_1 = \lambda_2 k_C, (1-\lambda) k_C, \qquad \lambda_1 = \text{mass fraction of product}$$

$$R_2 = (1-\lambda_1-\lambda_2)(k_I + \lambda_2 k_B) - \lambda_2 k_C, \quad \lambda_2 = \text{mass fraction of radical}$$

Ideal equation of state:

$$e = \frac{p}{(\gamma-1)\rho} + \lambda_1 Q_1 + \lambda_2 Q_2$$

where

$$Q_1, Q_2 = \text{heat release to form product } (< 0) \text{ and radical } (> 0)$$

Ignition and Growth reaction model (Lee and Tarver, 1980)

$$\mathcal{F}(solid) \longrightarrow \mathcal{P}(gas)$$

Reaction rate: (hot spots)

$$R = k_I + k_{G_1} + k_{G_2}$$

where

$$\begin{aligned}
k_I &= I(1-\lambda)^b\ (\max\{\rho - 1 - a, 0\})^x && \text{if } \lambda < \lambda_I && \text{(ignition)}\\
k_{G_1} &= G_1(1-\lambda)^c \lambda^d p^y && \text{if } \lambda < \lambda_{G_1} && \text{(rapid growth)}\\
k_{G_2} &= G_2(1-\lambda)^e \lambda^g p^z && \text{if } \lambda > \lambda_{G_2} && \text{(slow growth)}
\end{aligned}$$

Mixture JWL equation of state:

$$e = (1-\lambda)e_s + \lambda e_g, \qquad \frac{1}{\rho} = (1-\lambda)v_s + \lambda v_g$$

where

$$\begin{aligned}
e_s &= \frac{p_s v_s}{\omega_s} - F_s(v_s) + Q \qquad & e_s &= C_s T_s + G_s(v_s) + Q\\
e_g &= \frac{p_g v_g}{\omega_g} - F_g(v_g) \qquad & e_g &= C_g T_g + G_g(v_g)
\end{aligned}$$

and

Closure conditions: $p_s = p_g$ and $T_s = T_g$

Detonation Formation

Geometry: quarter plane $x > 0$, $y > 0$

Reaction/EOS: one-step/ideal with

$$\gamma = 1.4, \quad T_C = 1, \quad \epsilon = .075, \quad Q = -4, \quad \sigma = \frac{\epsilon}{(\gamma - 1)Q}$$

Initial conditions: prescribed temperature gradient

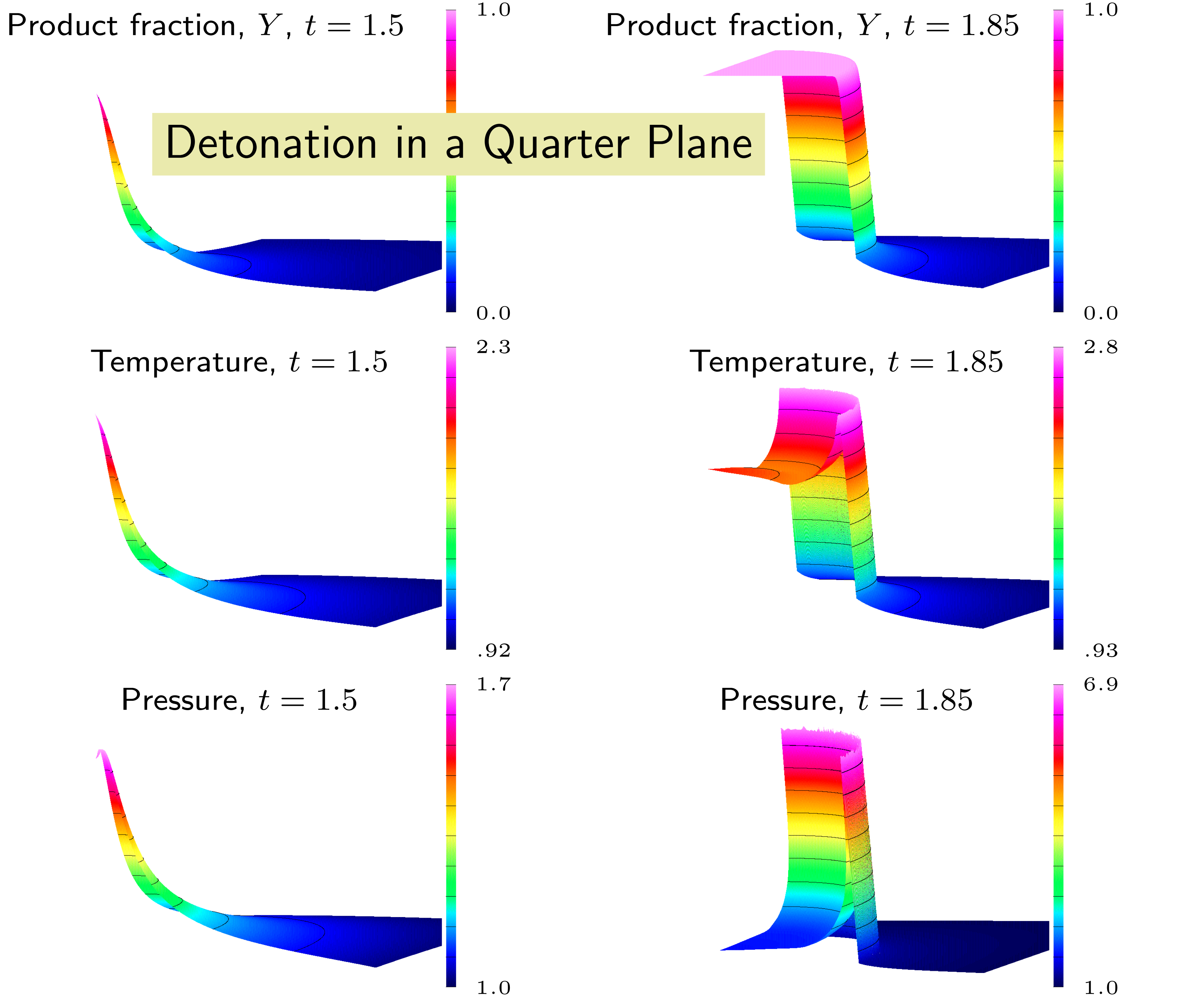
$$u = v = \lambda = 0, \quad p = 1, \quad T = 1 - \delta\sqrt{x^2 + y^2}, \quad \delta = .0375$$

Boundary conditions: solid walls on $x = 0$, $y = 0$

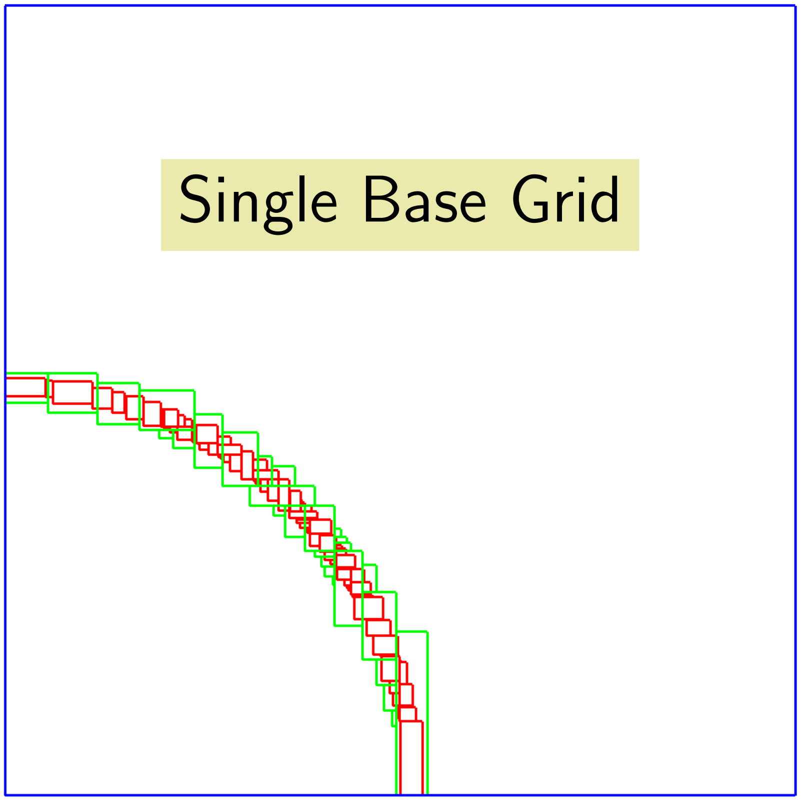
Base grid: 400×400 grid cells for the domain $0 < x < 2$, $0 < y < 2$

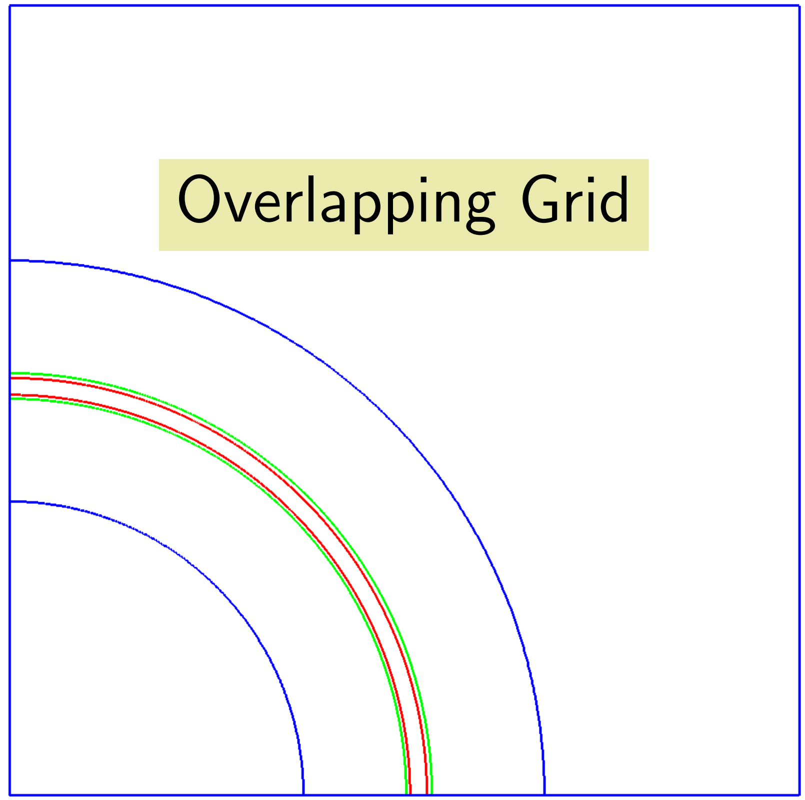
AMR: 2 child grid levels, refinement factor=4

Detonation in a Quarter Plane

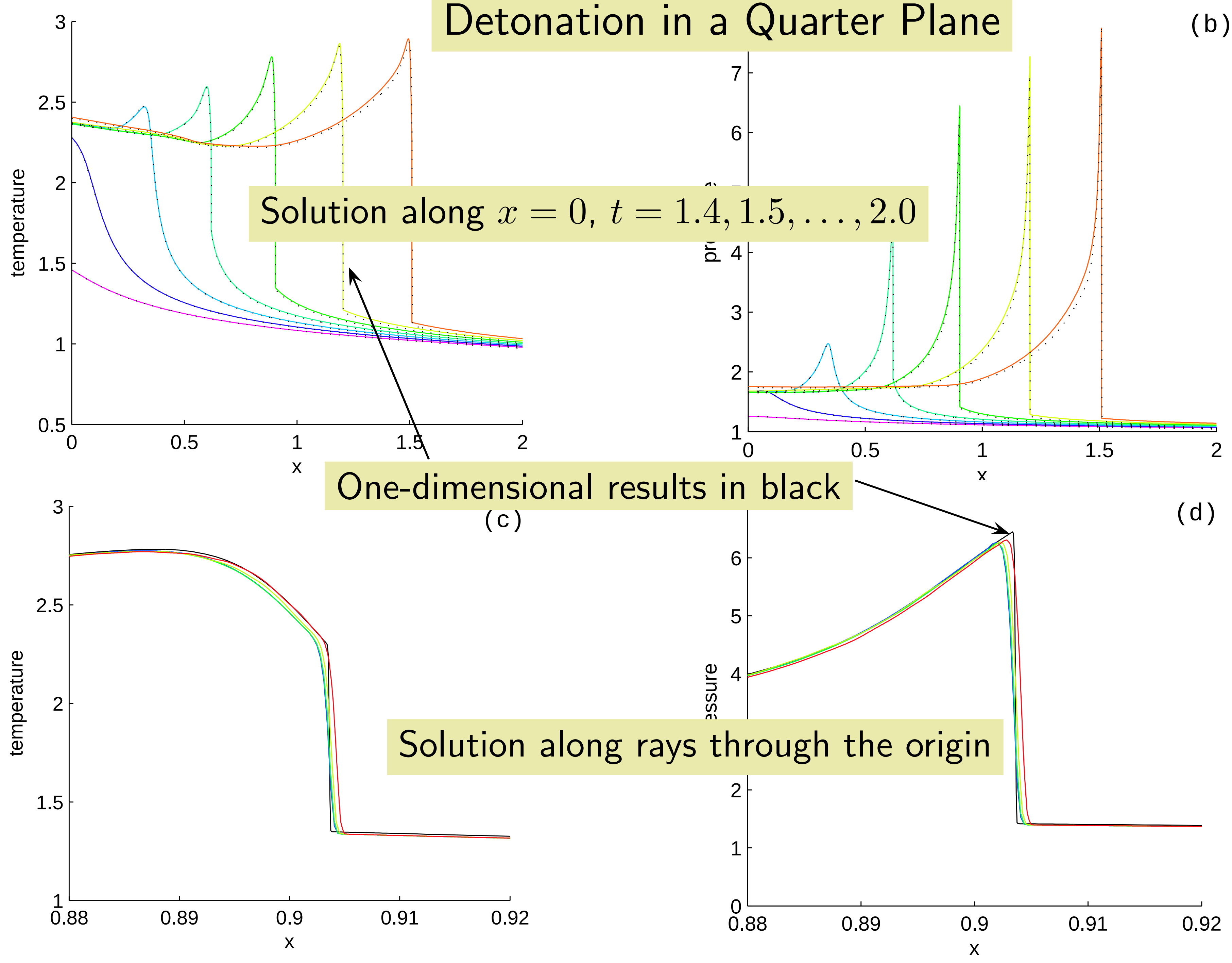


Detonation in a Quarter Plane





Detonation in a Quarter Plane



Detonation Entering the Annulus Grid

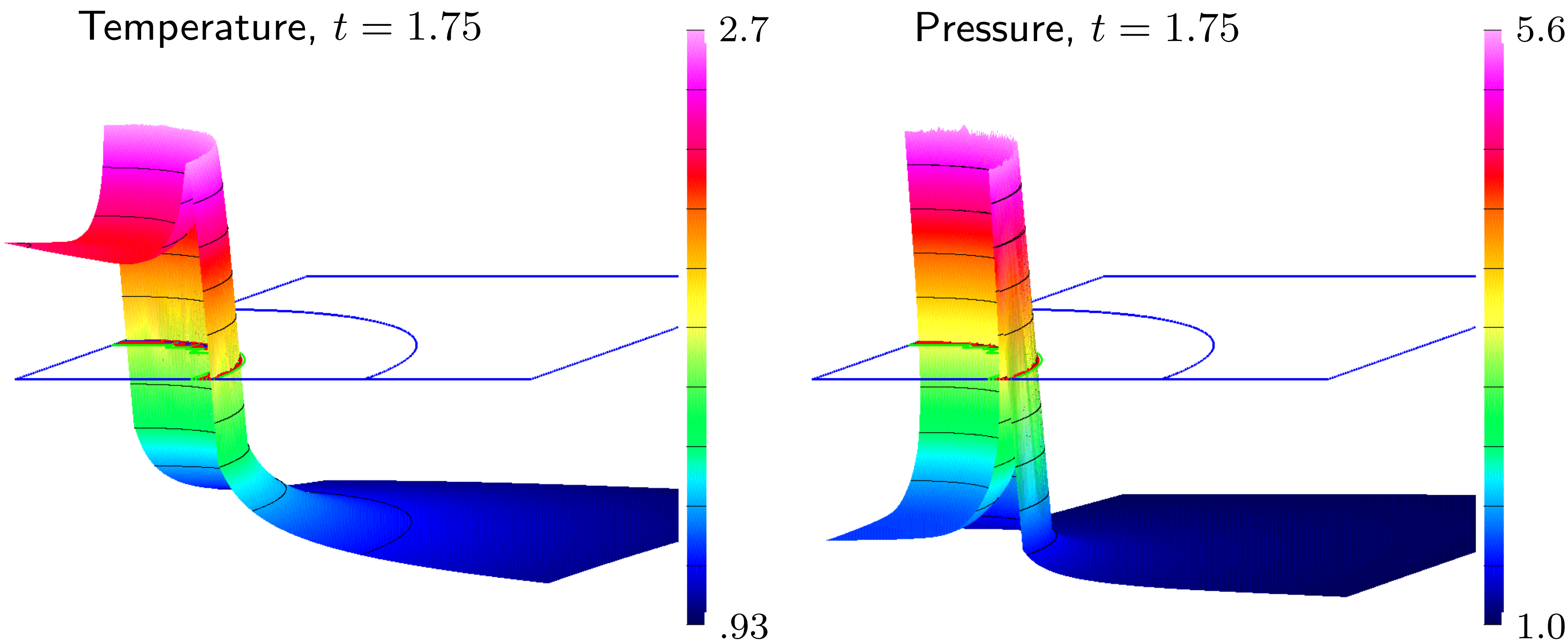


Figure 1: Behavior of the temperature and the AMR grid at $t = 1.75$ for a two-dimensional calculation on a rectangular base grid with an embedded annular grid.

Overlapping Grid vs Single Base Grid Results

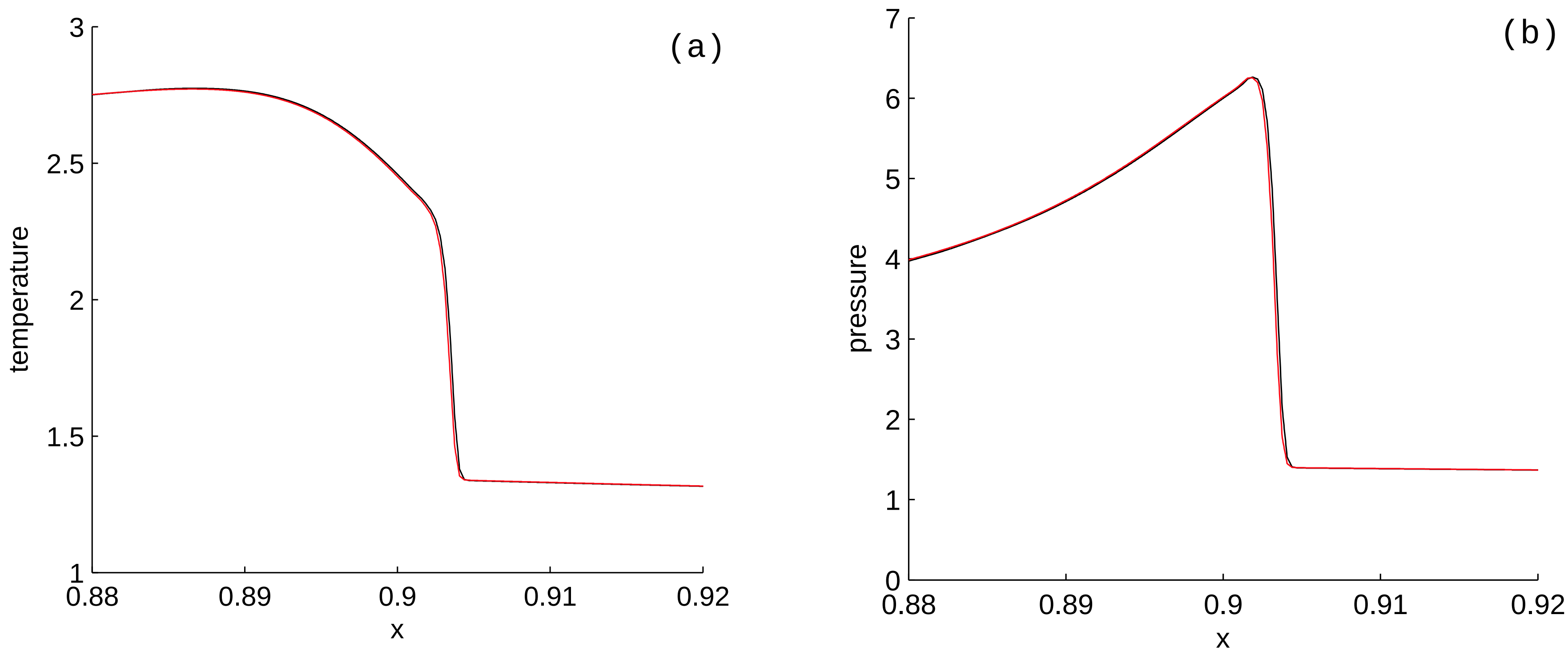
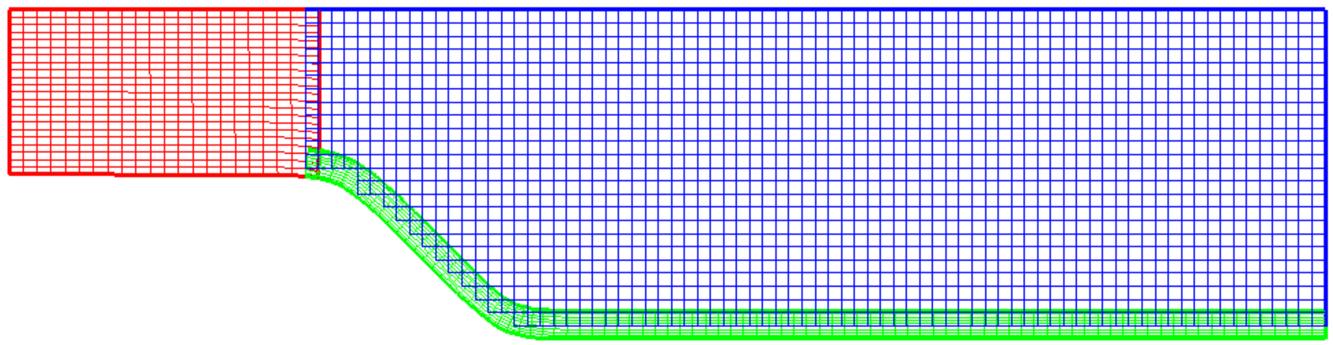


Figure 2: Behavior of the temperature (a) and pressure (b) along $y = 0$ in the vicinity of the detonation at $t = 1.8$. The red curves are from the grid with the embedded annulus and the black curves are from the rectangular grid with no annulus.

Detonation Diffraction

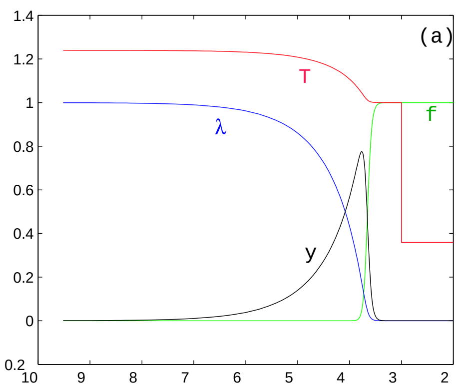
Geometry: smooth expanding channel

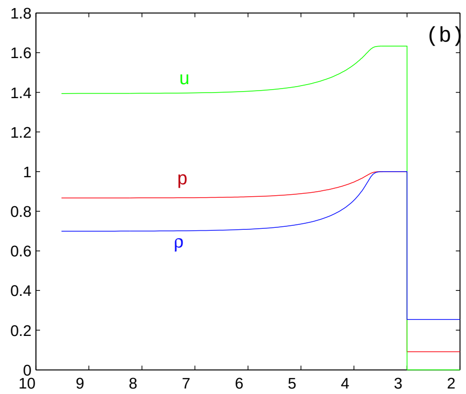


Reaction/EOS: chain-branching/ideal with

$$T_I = 3, \quad T_B = 0.75, \quad \epsilon_I = 0.05, \quad \epsilon_B = 0.125, \quad \gamma = 1.4, \quad Q_1 = -1, \quad Q_2 = 0$$

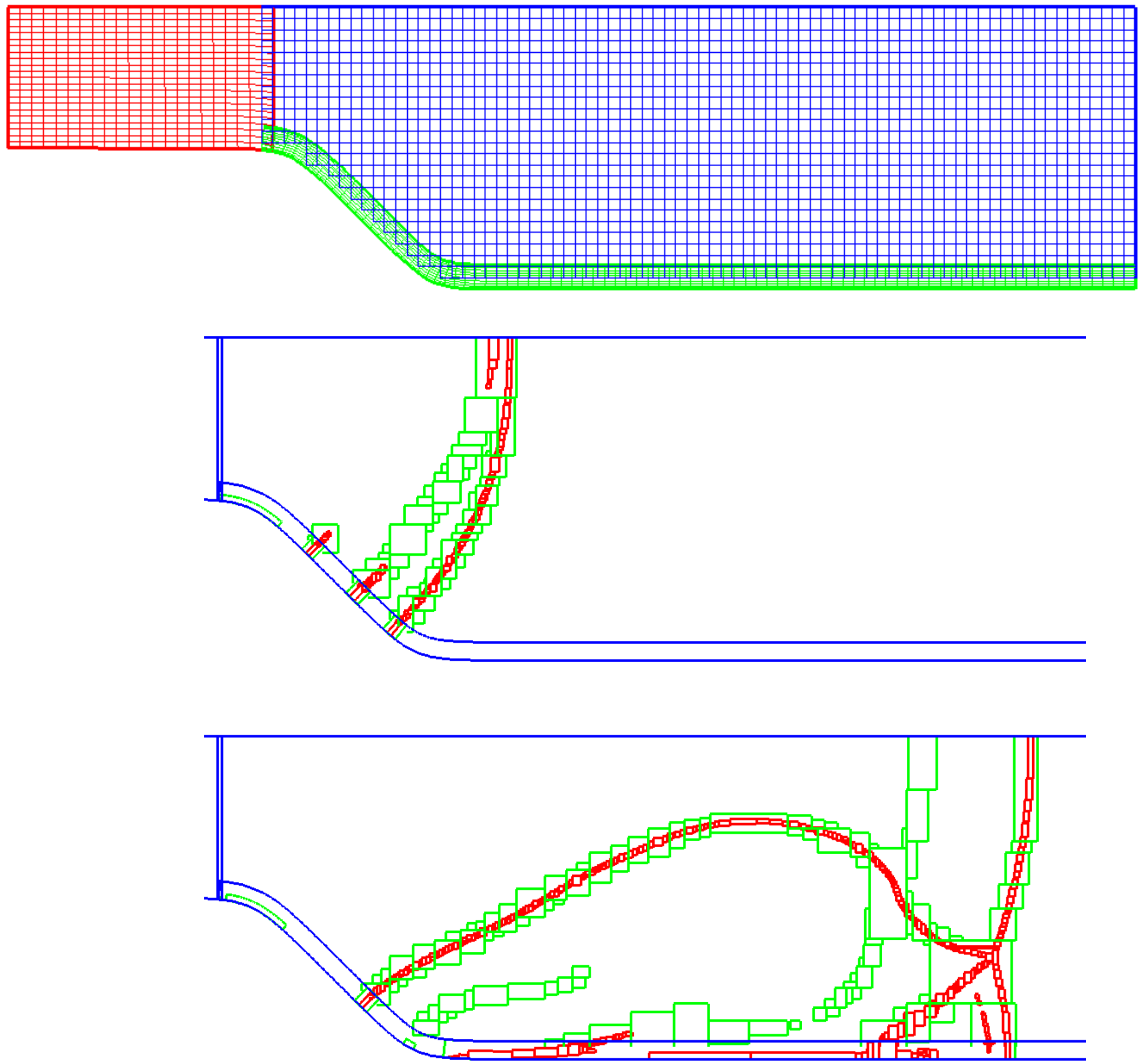
Initial conditions: steady over-driven detonation.

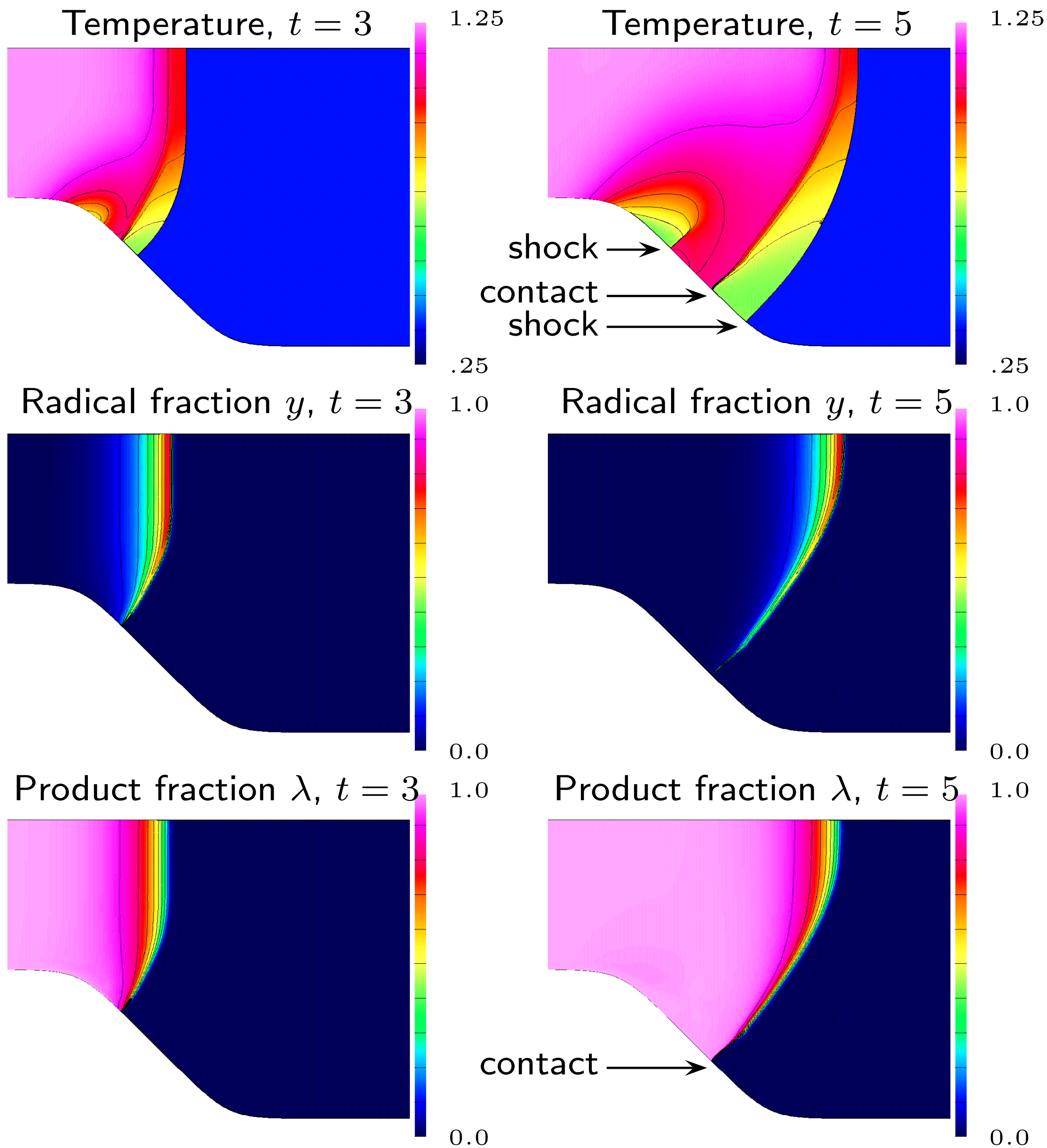


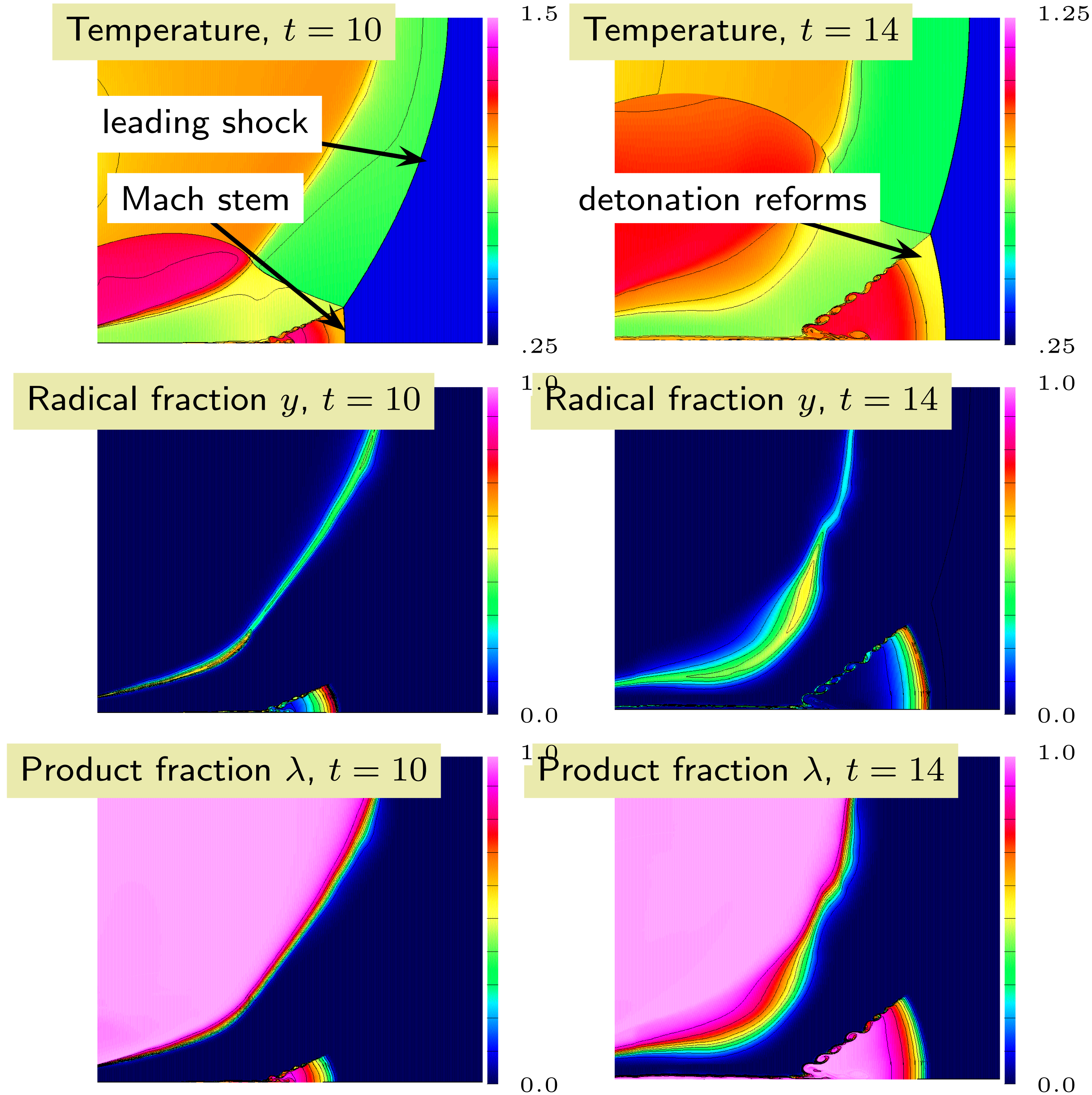


AMR: 2 child grid levels, refinement factor=4

Detonation in an Expanding Channel.




Temperature, $t = 3$
1.25
.25
Temperature, $t = 5$
1.25
.25
shock
contact
shock
Radical fraction y, $t = 3$
1.0
0.0
Radical fraction y, $t = 5$
1.0
0.0
Product fraction λ, $t = 3$
1.0
0.0
Product fraction λ, $t = 5$
1.0
0.0
contact


Temperature, $t = 10$
1.5
.25
leading shock
Mach stem
Temperature, $t = 14$
1.25
.25
detonation reforms
Radical fraction y, $t = 10$
1.0
0.0
Radical fraction y, $t = 14$
1.0
0.0
Product fraction λ, $t = 10$
1.0
0.0
Product fraction λ, $t = 14$
1.0
0.0

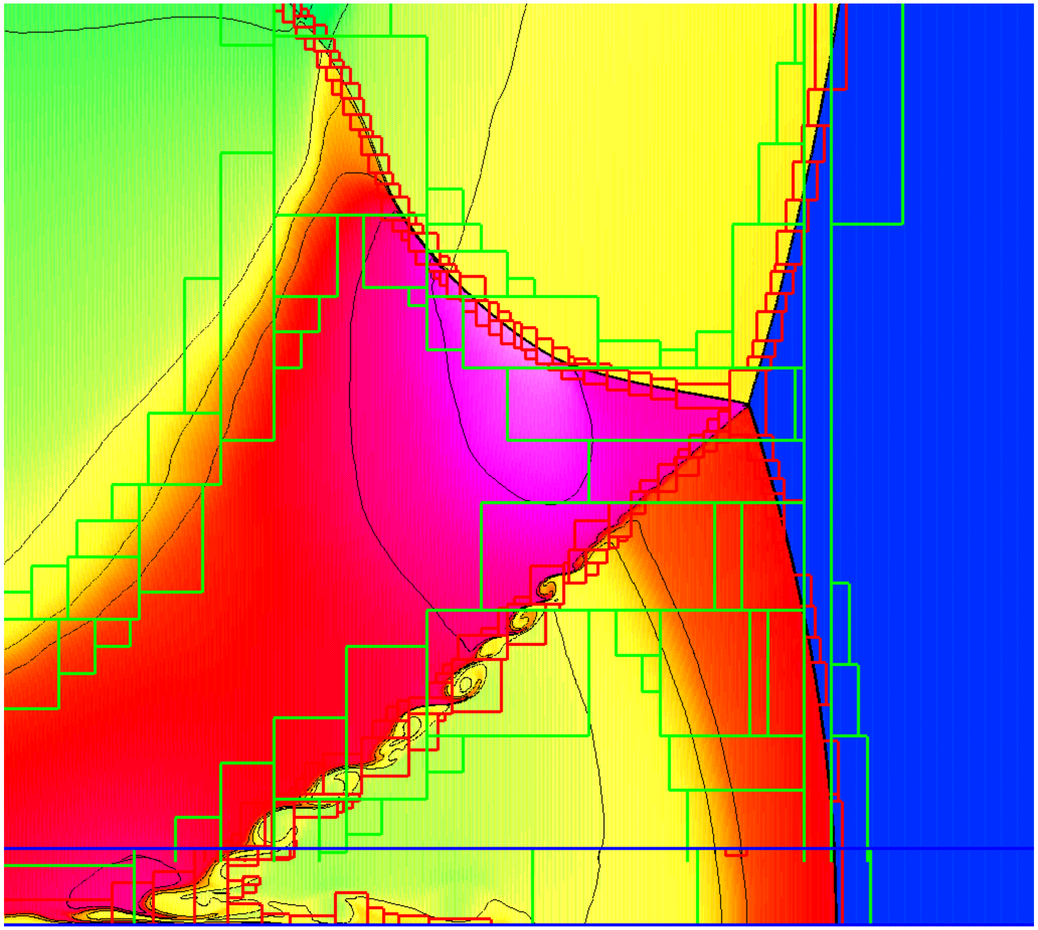


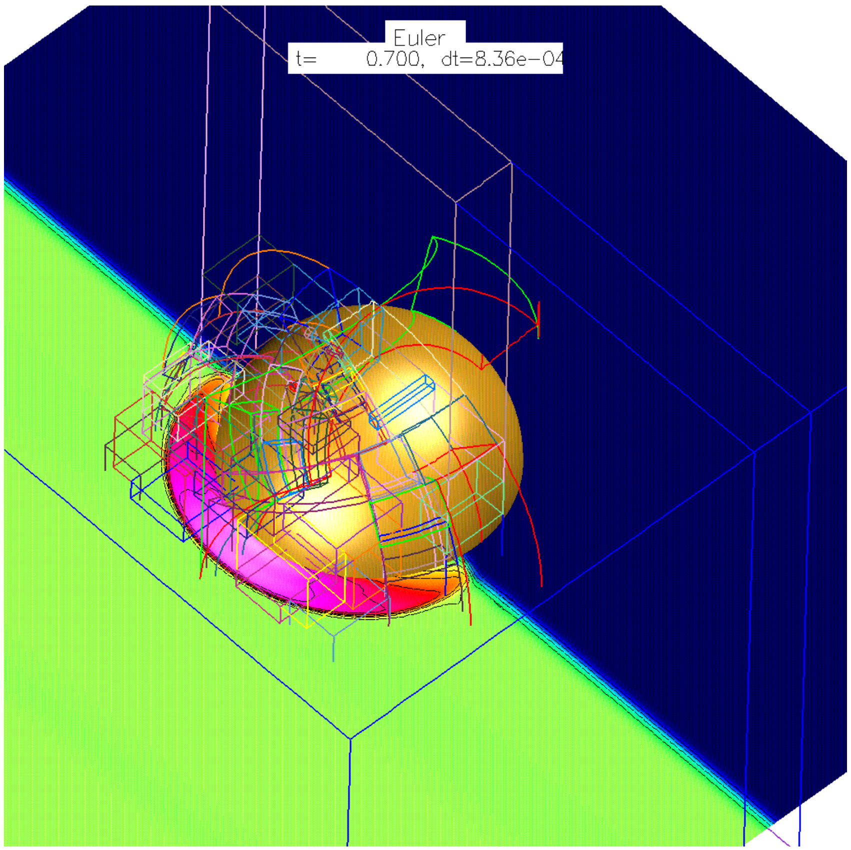
Figure 3: Closeup of the density near the Mach stem. The boundaries of the refinement grids are shown.

AMR performance on two detonation problems:

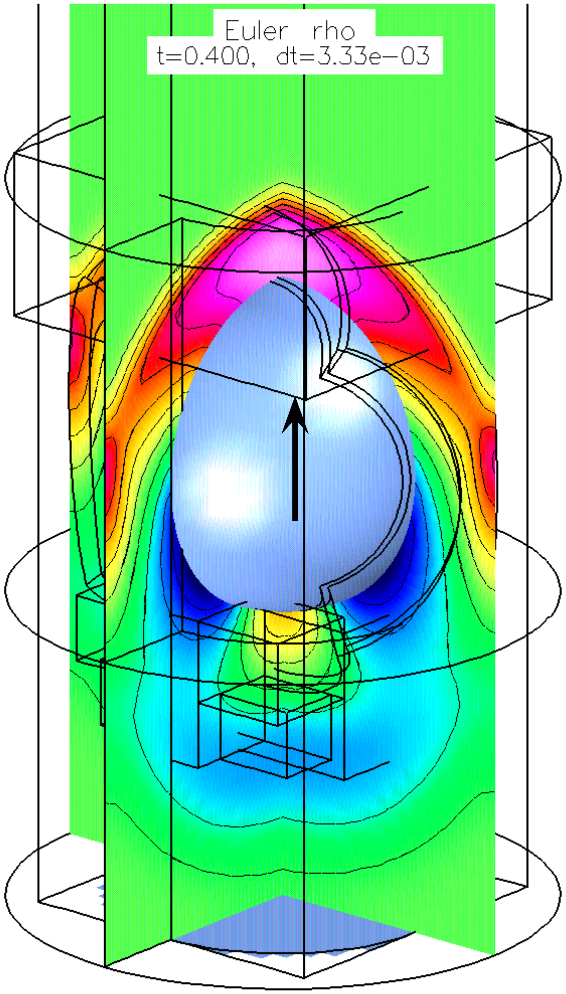
	Quarter plane		Expanding channel	
time steps	$12,418$		$21,030$	
grids (min,ave,max)	$(2,57,353)$		$(5,274,588)$	
points (min,ave,max)	$(2.0e5, 9.2e5, 1.9e6)$		$(1.2e5, 6.4e5, 1.3e6)$	
	s/step	%	s/step	%
compute $\Delta u_{i,j}^n$	13.85	92.7	11.50	82.4
boundary conditions	.12	.8	.14	1.0
interpolation (overlapping)	.09	.6	.45	3.2
AMR regrid/interpolation	.54	3.6	1.62	11.6
other	.34	2.3	.25	1.8
total	14.94	100	13.96	100

Table 1: CPU time (in seconds) per step for various parts of the code and their percentage of the total CPU time per step.

Current work: three-dimensions



Shock hitting a sphere (Euler)



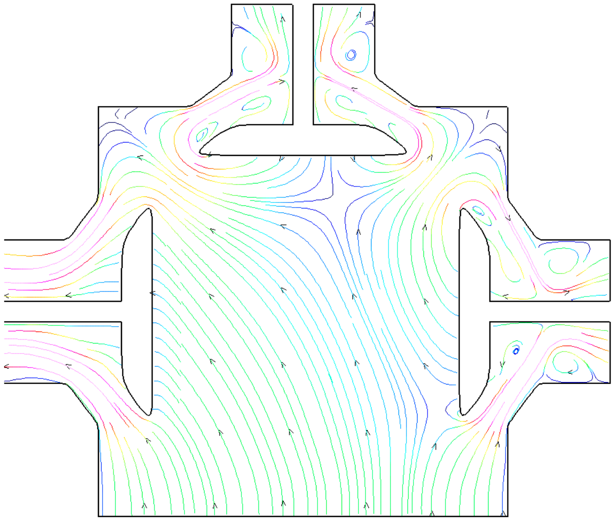
Sphere moving in a tube
with AMR (Euler)

Current work: moving geometry

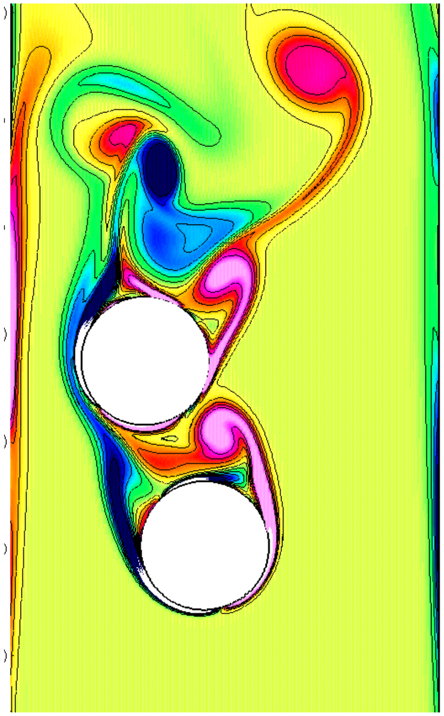
◇ The governing equations are written in the moving coordinate system

◇ Support for (1) specified motion, (2) rigid-body motion with forces and torques determined from the flow.

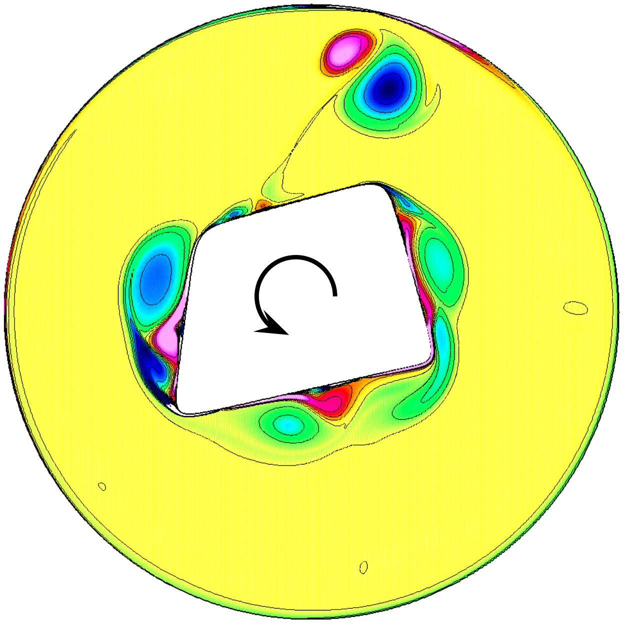
◇ The grids are moved at every time step and the interpolation points are recomputed.



Moving valves (INS)



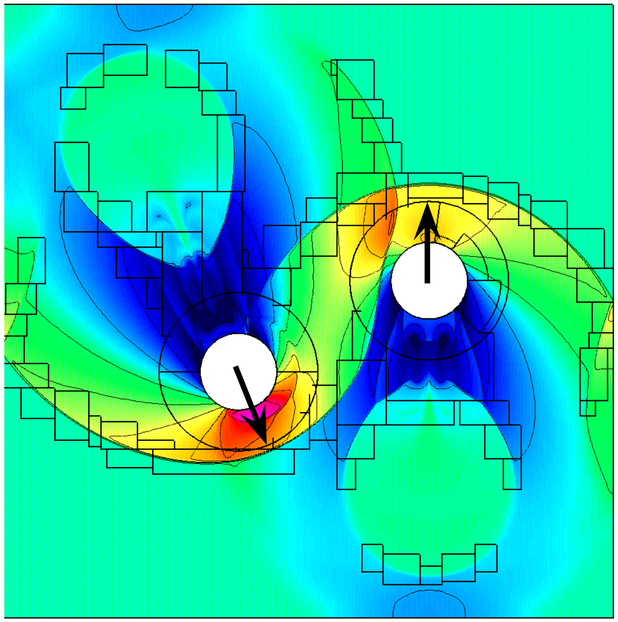
Falling cylinders (INS)



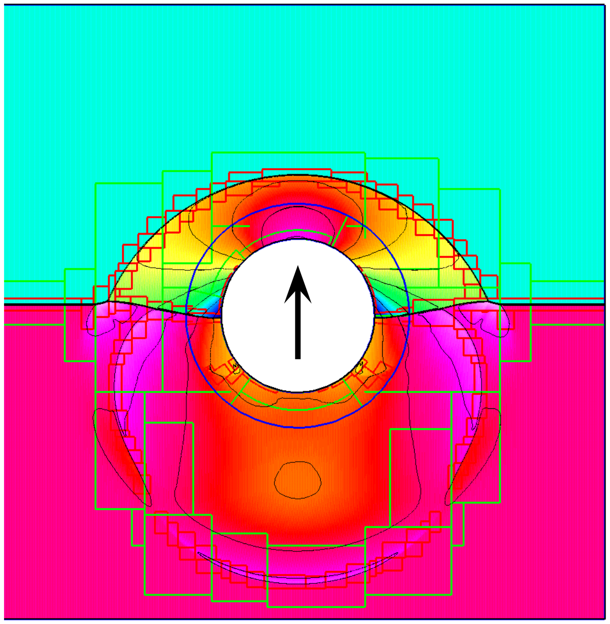
Rotating body (INS)

Current work: moving geometry and AMR

- Refinement grids move with the corresponding base grid.
- The AMR regrid step is performed at the start of the step, followed by the grid movement.



Moving cylinders (Euler)



Cylinder moved by a shock (Euler)

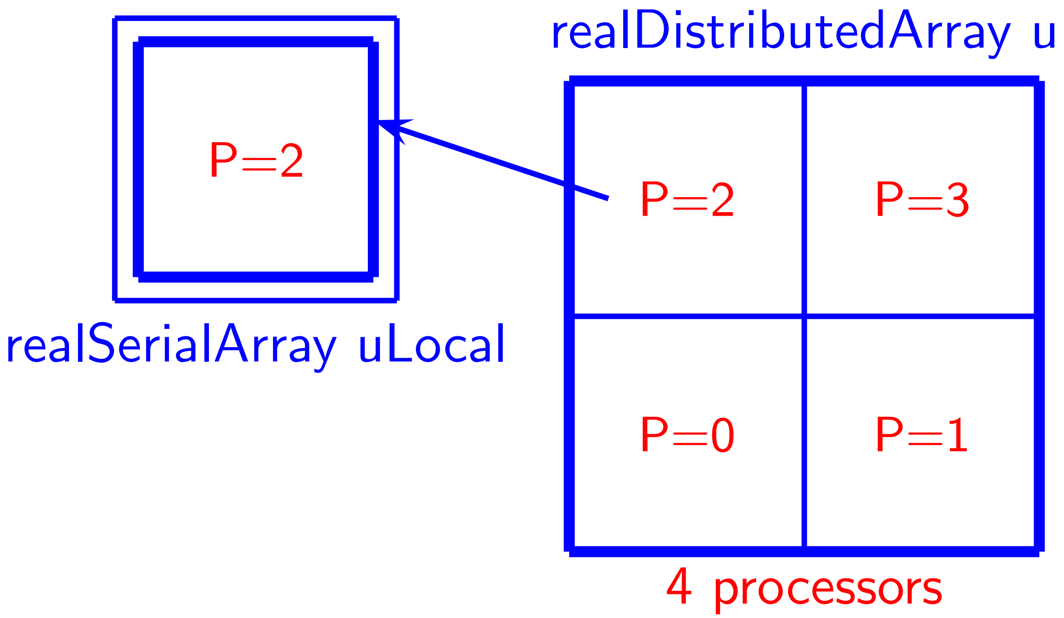
Current work: A parallel version of the flow solver OverBlown

- ♢ Grids can be distributed across one or more processors.
- ♢ Distributed parallel arrays using P++ (K. Brislawn, B. Miller, D. Quinlan)
- ♢ P++ uses Multiblock PARTI (A. Sussman, G. Agrawal, J. Saltz) for block structured communication with MPI (ghost boundary updates, copies between different distributed arrays)
- ♢ A special parallel overlapping grid interpolation routine has been written.

NP	sec/step	ratio
1	8.4	1.
2	4.3	2.

Table 2: Shock hitting a cylinder (no AMR), 1.1 million grid-points, Dell workstation 2.2GHz Xeon.

P++ : parallel multi-dimensional arrays



Partitioning_Type partition; // object that defines the parallel distribution

partition.SpecifyInternalGhostBoundaryWidths(1,1);

realDistributedArray u(100,100,partition); // build a distributed array

u=5.;

realSerialArray & uLocal = u.getLocalArray(); // access the local array

myFortranRoutine(*uLocal.getDataPointer(),...);

u.updateGhostBoundaries();

Overture and OverBlown are available for download,
www.llnl.gov/CASC/Overture.html